FRAMEWORK FOR ROBOTS SELF-DEPLOYMENT USING VIRTUAL FORCE APPROACH

BY

Gamal Ameen Saeed Sallam

A Thesis Presented to the
DEANSHIP OF GRADUATE STUDIES

KING FAHD UNIVERSITY OF PETROLEUM & MINERALS

DHAHRAN, SAUDI ARABIA

In Partial Fulfillment of the
Requirements for the Degree of

MASTER OF SCIENCE

In

COMPUTER NETWORKS

APRIL 2016

KING FAHD UNIVERSITY OF PETROLEUM & MINERALS
DHAHRAN 31261, SAUDI ARABIA

DEANSHIP OF GRADUATE STUDIES

This thesis, written by **GAMAL AMEEN SAEED SALLAM** under the direction of his thesis adviser and approved by his thesis committee, has been presented to and accepted by the Dean of Graduate Studies, in partial fulfillment of the requirements for the degree of **MASTER OF SCIENCE IN COMPUTER NETWORKS**.

Thesis Committee

U. Baroudi

Dr. Uthman Baroudi (Adviser)

(Co-adviser)

Ashraf

Dr. Ashraf Mahmoud (Member)

Dr. Sami El Ferik (Member)

(Member)

Dr. Ahmad S. Almulhem
Department Chairman

Dr. Salam A. Zummo
Dean of Graduate Studies

19/5/16
Date

عمادة الدراسات العليا
DEANSHIP OF GRADUATE STUDIES

Dedicated to my family members

ACKNOWLEDGMENTS

All praises, all glory and all thanks are due to Allah, The Majestic, The Almighty for bestowing me with knowledge, guidance, patience, courage and health to achieve this work. May peace and blessings be upon prophet Mohammed (PBUH), his family and his companions. I would like to acknowledge the King Fahd University of Petroleum & Minerals and the Ministry of Higher Education in Yemen for the support extended towards my research and providing me the opportunity to pursue graduate studies. I would like to express my deep thanks to Dr. Uthman Baroudi, my thesis advisor, for the enormous support he provided me throughout my thesis work. I don't have enough words to express my gratitude for his guidance, support, and motivations. He was my source of inspiration, mentor, and friend at KFUPM. I am also grateful for Dr. Ashraf Mahmoud for his feedback on my thesis work and the courses he taught me where I learned the basics of how to do research. Also, I would like to thank Dr. Sami El-Ferik for his constructive comments on my work. I can't forget my friends at KFUPM, all of them were in a great support to me. Special thanks to my labmate and roommate Mohammed Alshabot, our discussions were always helpful and fruitful. I also want to thank Mustafa, Al-Areeq, Taher, Al-Daroubi, Yousef,

they are more than friends. My ultimate gratitude and love for my parents, brothers, sister, and my wife. Without their endless support, I would not be able to start and finish my study at KFUPM. I can't thank my parents enough here, may Almighty Allah keep them healthy and give me the opportunity to serve them throughout my life. For my wife, you were very supportive and patience especially being away from each other most of the time during my thesis work.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

THESIS ABSTRACT

NAME:	Gamal Ameen Saeed Sallam
TITLE OF STUDY:	Framework for Robots Self-Deployment Using Virtual Force Approach
MAJOR FIELD:	Computer Networks
DATE OF DEGREE:	April 2016

Robots deployment is becoming so popular in recent days due to its applications in different fields. Robots are used for many military and civilian applications. Many such applications, such as search-and-rescue operations or area monitoring during an environmental disaster, cannot be effectively carried out by a single robot, but rather are carried out by many robots linked cooperatively in a robotic network. In rescue operations, for example, robots can be used to help in discovering bodies under the rubbles or even assist the injured. One of the main challenges in these applications is how to deploy the robots without central coordination. Virtual force (VF) technique appears as one of the prominent approaches to perform multi-robot deployment autonomously. In this thesis, we propose a unified framework that is generalized to consider different aspects of VF-based deployment. First, the ef-

fectiveness of virtual force depends on how its parameters are selected in order to achieve the required deployment. There are two important factors: attractive force (w_a) and repulsive force (w_r). We investigate how to calibrate these two factors in order to accommodate different kinds of scenarios with different number of robots, communication ranges and so on. Moreover, we present an energy aware virtual force approach to balance energy consumption among robots and consequently maximize the network lifetime. Second, since most of the existing VF-based approaches lack purposeful deployment, we present a Cooperative Virtual Force Robot Deployment (COVER) technique. Our approach modifies the original VF approach to consider the mission requirements such as the number of required robots in each locality (e.g., landmarks are distributed and each needs a specific number of robots in its vicinity). In addition, COVER expedites the deployment process by establishing a cooperative relation between robots and neighboring landmarks. To shorten the deployment time and improve other metrics we propose Two-hop COVER that enhances COVER in many different ways. In case that Two-hop COVER could not reach 100% demand satisfaction, we present Trace Fingerprint that is designed to be used with Two-hop COVER to guarantee 100% demand satisfaction. Extensive simulation experiments have been carried out to assess the performance of the proposed algorithms. Moreover, a proof-of-concept experiment using Turtlebot robots has been carried out for validating COVER algorithm.

ملخص الرسالة

الاسم الكامل: جمال امين سعيد سلام

عنوان الرسالة: إطار لنشر الربوتات بإستخدام طريقة القوى الوهميه

التخصص: هندسة شبكات

تاريخ الدرجة العلمية: 2016

توزيع الربوتات اصبح مشهورا بالاونه الاخيره نتيجة للتطبيات اللي ستتيحها في مختلف المجالات. الربوتات تستخدم في العديد من التطبيقات سواء المدنيه او العسكريه. عدد من تلك التطبيقات مثل تطبيقات البحث والانقاذ او المراقبه خلال الكوارت البيئيه لا يمكن تنفذيها بربوت منفرد ولكنها تتطلب تعاون اكثر من ربوت متوصلين بإستخدام شبكة الربوتات. على سبيل المثال, في عملية الانقاذ يمكن استخدام الربوتات لإكتتشاف الأجسام تحت الانقاض او مساعدة الجرحى. احد اهم العوائق التي تواجه مثل هذي التطبيقات هو كيفية نشر الربوتات بدون تحكم مركزي. تعتبر طريقة القوى الوهميه واحده من اهم الطريق المستخدمه لنشر الربوتات بدون تحكم مركزي. في هذه الأطروحه, نقترح إطار موحد للأخذ بالإعتبار عدد من العوامل المؤثره في طريقة نشر الربوتات بإستخدام القوى الوهميه. اولا, فعلية هذه الطريقه تعتمد على طريقة ضبط عناصرها الأساسيين. يوجد عنصريين مهمين وهما قوة التجاذب وقوة التنافر. سنبحث في هذه الدراسه كيفية ضبط هذين العنصرين بحيث يسهل استخدام الطريقه لأكثر من سيناريو. علاوة على ذالك, سنعرض طريقة القوى الوهميه بحيث تأخذ بالاعتبار استهلاك الطاقه والطاقه المتبقيه لكل ربوت. ثانيا, بما أن معظم الطرق الحاليه التي تعمتمد على طريقة القوى الوهميه لا تأخذ بالاعتبار نشر الربوتات لغرض معين, فإننا نقترح في هذه الدراسه طريقة جديده تسمى نشر الربوتات بإستخدام القوى الوهميه التعاونيه. الطريقه المقترحه تعدل بطريقة القوى الوهميه لتأخذ بالإعتبار متطلبات المهمه مثل عدد الربوتات المطلوبه في كل منطقه من مناطق نشر الربوتات. علاوة على ذالك فإن الطريقه المقترحه تحسن عملية نشر الربوتات من اخلال استغلال التعاون بين الربوتات وبين علامات مميزه موجوده مسبقا في أماكن نشر الربوتات. طريقتين اخريين تم اقتراحهما لتحسين عملية توزيع الربوتات والوصول الى افضل نتيجه ممكنه. تم دراسة الطرق المقترحه بإستخدام المحاكاه وبإستخدم تجارب على روبوتات واقعيه وجميعهم اثبتو فعالية الطرق المقترحه.

CHAPTER 1

INTRODUCTION

1.1 Robots deployment

Robotics networks have gained a lot of attention in the last decade due to the major technological advances that widened the scope of their applications. A robot can be equipped with sensors, which enable such a network type to combine the advantages of its mobility and actuation features with the wireless sensor network capabilities. Robots deployment can be done manually, by placing each node in a pre-determined position in an area of interest such that some desired coverage and connectivity properties are ensured. Alternatively, robots may dynamically adjust their positions by self-spreading such that the covered area is maximized while maintaining the inter-robot connectivity. In harsh environments where human intervention is not possible, robots can be used for monitoring specific phenomena, reporting data to a base-station and taking action as deemed feasible and appropriate. The reactive nature of robots and the applications in which a robotics network is effective make deterministic placement impractical and adaptive self-deployment of robots more appropriate. A number of algorithms has been proposed to address adaptive self-deployment as in [1–7]. One of the most popular techniques to enable robots self-spreading after an ad-hoc random placement in an area is to model them as electromagnetic particles that exert virtual forces where robots repel or attract neighbors based on proximity [1] [2]. Based on the composite force applied by its neighbors, a robot moves to a new location. This process is repeated many times until the network reaches equilibrium status where robots become uniformly distributed in the area. It has the following advantages: a simple communication model (size and type of the packets), enhancement of the initial coverage degree, the control of the coverage degree by the threshold value, fast convergence, the consideration of obstacles, borders, and coverage holes. However, The efficiency of this algorithm depends on the parameters attractive force w_a and repulsive force w_r [8]. Other approaches lie in one of the following classes: computational geometry based, fuzzy based, and metaheuristic based [8]. In the computational geometry based approach [4, 9, 10], a geometric computation is used to find out the places that are not covered yet and direct robots in the densely areas to move toward them. Voronoi diagram and Delaunay triangulation are two common approaches in this class. The weaknesses of this approach are: it is a greedy algorithm, and it is ineffective when dealing with large holes [11]. In the fuzzy based approach [12, 13], fuzzy logic system is used to control the robot movement. The fuzzy system puts several rules based on the Euclidean distance or the number of robots for example. Then the system will provide a new position that each robot should relocate to it. It does not take into account the presence of obstacles. Algorithms belonging to metaheuristic based approach utilizes the effectiveness of metaheuristics in order to settle the position, the direction, and the movement speed of a mobile sensor. Ant Colony (AC) [14], Genetic Algorithms (GA) [15], Particle Swarm Optimization (PSO) [16], and Simulated Annealing (SA) [8] are examples of such algorithms. It has a high complexity and the quality of the obtained solutions depends on a large number of parameters (e.g., number of iterations and GA-related parameters) [8].

1.2 Virtual Force Approach (VF)

One of the most popular approaches for robots deployment is the virtual force (VF). VF-based deployment is inspired by the artificial potential field-based techniques in the field of robotic obstacle-avoidance [17], in which nodes are treated as virtual particles and the virtual forces due to potential fields repel the nodes and the obstacles. In this approach, there are three kinds of objects: robots, obstacles and preferential areas, all of them exert a different kind of force, either an attractive force or a repulsive force. Robots exert a repulsive force or an attractive force toward each other depending on the Euclidean distance between them, while obstacles exert a repulsive force and preferential areas exert an attractive force on robots. In this approach, we consider the following assumptions. First, each robot has a sensing range R_s and a communication range $R_c = K.R_s$ and $k > 1$.

Within the sensing range, the robot can detect the local environment conditions or implement a certain task. Moreover, a robot can communicate with other robots within its communication range. Second, the positions of all the robots are known and can be acquired using any localization technique or using GPS. Third, all the robots can move according to the calculated results of the algorithm. Fourth, the communication range is greater than the sensing range.

1.3 Robots Movement of the Virtual Force

In the virtual force model, obstacles are assumed to exert repulsive (negative) forces on a robot. However, areas of preferential coverage exert attractive (positive) forces on a robot. Let F_{iA} be the total (attractive) force on robot R_i due to preferential coverage areas, and let F_{iR} be the total (repulsive) force on robot R_i due to obstacles. The total force F_i on R_i can now be expressed as in Equation 4.1.

$$F_i = \sum_{j=1,ji}^{k} F_{ij} + F_{iR} + F_{iA} \tag{1.1}$$

where

$$F_{ij} = \begin{cases} w_a(d_{ij} - d_{th}), \theta_{ij} & \text{if } d_{ij} > d_{th} \\ 0 & \text{if } d_{ij} = d_{th} \\ \frac{w_r}{d_{ij}}, \pi + \theta_{ij} & \text{if } d_{ij} < d_{th} \end{cases} \tag{1.2}$$

and

$$F_{iA} = w_a * (d_{iA}), \theta_{iA} \tag{1.3}$$

$$F_{iR} = \frac{w_r}{d_{iR}}, \theta_{iR} + \pi \tag{1.4}$$

$$x_i' = x_i + \vec{F}_{xi} \tag{1.5}$$

$$y_i' = y_i + \vec{F}_{yi} \tag{1.6}$$

w_a is the attractive force, w_r is the repulsive force, and d_{ij} is the distance between robot i and robot j, d_{th} is the distance threshold between any two nodes which control how close will be the nodes from each other, and θ_{ij} is the angle between node i and node j. d_{iA} is the distance between robot i and preferred point A, d_{iR} is the distance between robot i and obstacle R.

1.4 Motivations

As we have stated earlier, robots deployment has a lot of applications in different fields. In order for the deployment to be effective for different kind of scenarios such as in hazardous environments, the deployment should be carried out autonomously without any central coordination. Virtual force appears as one of the prominent approaches for this purpose. However, this approach has different problems that we are going to address in the following study. The setting of virtual force parameters, specifically the attractive force w_a and the repulsive force w_r has not been discussed nor mentioned accurately in the literature. In [1], they just mentioned that the attractive force should be highly smaller than the repulsive force. Therefore, in this thesis, the goal is to formulate these two factors to be a function of system parameters such as the number of robots and communication threshold, etc. Also, considering the remaining energy in each robot is another important factor to be considered to maximize the lifetime of the network. Some robots may start to deplete their energy earlier than other robots so the possibility of these robots to shut down is very high and could cause a disruption in the network. So, in this work we present an energy aware virtual force that considers the heterogeneous level of energy among robots such that the ones with a low level of energy will be able to switch to a power save mode and consequently conserve the remaining energy as possible.

Moreover, Most of the previous works have focused on how to maximize the covered area and how to achieve a uniform distribution of the robots [1–7,18]. But some scenarios may require that some places should have more robots compared to others. More specifically, each place could have a level of emergency different from other places and it would ask for a specific number of robots. To illustrate, assume that one or more landmarks are deployed in the area of interest. The landmark is a devise that is equipped with special capabilities, e.g., sensing and computational resources, to enable them to assess the situation in their vicinity. These landmarks are deployed in order to support the deployment of the mobile robots when an event takes place. The landmarks monitor the area, and determine a need for the mobile robots in that area after an incident has occurred. The landmarks may be equipped with whatever types of sensors or detectors that are appropriate for their function, and are not restricted to any particular type of detection mode. For example, the landmarks may be equipped with chemical sensors to analyze air, water quality or a gas, liquid, or vapor concentration, toxic gas detectors, water level detectors, seismometers, visibility meters, or any other sensors which provide data from which a need for mobile robots can be determined. A demand, or need, for robots is determined by the landmark from its monitoring of the area around it and a predetermined formula using the results of the monitoring. A number of robots already available to the landmark, if any, will be subtracted from the number of robots calculated from the predetermined formula. For example, in an aspect of the work directed to a search-and-rescue scenario at sea, readings such as a current strength and a wave height can be included in the predetermined formula. In an aspect of the work directed to a search-and-rescue scenario due to a gas leak or environmental contamination, a concentration of a chemical or contaminant can be included in the predetermined formula, with a greater concentration of chemical or contaminant indicating a greater need for mobile robots. In an aspect of the work directed to an earthquake scenario, a Richter scale reading can be included in the predetermined formula, with a greater reading on the Richter scale indicating a greater need for robots. In all the previous scenarios, a landmark will request a specific number of robots to come to its vicinity. The robots will be dropped in any point in the area of interest. Then, those robots need a mechanism to organize themselves such that the demand of each landmark is met. In this work, we are going to address a dynamic coverage of each landmark based on cooperative landmarks and robots. Virtual force has been used for the general deployment of robots where the focus was to improve the coverage ratio, however, in this work, we propose a cooperative virtual force for implementing virtual force technique considering specific requirements depending on the deployment objectives.

Finally, in order to realistically realize a proposed solution, the physical properties of robots and the environment should be considered during the implementation. First, we consider the localization of the robots. In a harsh environment, the only convenient way is to use the odometer properties of the robot to induce its location. This approach has its error that accumulates with time. Some of the studies assume that localization is already achieved using GPS [19], however, GPS is not available in indoor places or in the areas that are covered by trees.

Additionally, GPS has an error that cannot be tolerated in small-scale scenarios. Another work [20] assumes that a localization method is available and so they build their work based on that assumption, however in a real implementation, localization is a major issue and its error is sometimes very high and consequently affects the overall performance. Another issue is obstacles presence which can interfere with robots movement during the deployment. Dealing with obstacles is not as simple as assumed by some studies. The presence of obstacles means a high processing of the feedback coming from the distance sensors as well as increasing the path length to the goal. Many previous works have treated obstacles as a repulsive force [1, 19, 21], however in our scenario we have a goal that the robot needs to go to (which is the landmark) so the robot may need to move around the obstacle to get back to the path of the goal. More importantly, even if there are no obstacles in the area of interest, the robots themselves could become obstacles for each other.

1.5 Thesis Contribution

The goal of this thesis is to propose a unified framework that covers different and general aspects of VF-based deployment. The contributions can be categorized in the following points:

1. We propose two calibrates one for the attractive force and one for the repulsive force which factor system parameters such as the number of robots while computing the attractive/repulsive forces. Moreover, we consider an energy

aware implementation for virtual force in order to balance the consumption of energy among robots with heterogeneous levels of the remaining energy.

2. We propose a novel cooperative and distributed method (COVER) for multi-robot deployment using virtual force based on landmarks demand. COVER approach is cooperative such that landmarks and robots will help each other to maximize and expedite the demand satisfaction of each landmark. Virtual force purpose is to maximize the covered area but without any guidance on how to achieve a purposeful deployment. Its way of searching for the demanding node is random. So, we aim to reduce the randomness by guiding VF approach to perform certain deployment requirements.

3. Moreover, we improved the performance of COVER by proposing Two-hop COVER that aims to shorten the deployment time and increase the level of demand satisfaction.

4. Finally, in order to guarantee 100% demand satisfaction, we propose Trace Fingerprint to be used with Two-hop COVER to reach the maximum possible level of demand satisfaction. Fairness in distributing the robots among landmarks in case that the number of robots is less than the collective demand is considered for COVER and Two-hop COVER as well.

5. We study the effects of a real implementation of the virtual force by considering the physical properties of the robots and obstacles and their overall effect on the deployment. Two variants of the virtual force approach, full

virtual force, and the original virtual force, are implemented taking in consideration all the physical aspects of Khepera III robot. Moreover, Turtlebot robots were used for testing the proposed COVER approach on a real robot.

1.6 Thesis Structure

The remainder of the thesis is organized as follows. Chapter 2 presents some related works. Virtual force parameters calibration and energy aware version of virtual force are presented in chapter 3. Chapter 4 presents the cooperative virtual force (COVER), Two-hop COVER and Trace Fingerprint. Finally, in chapter 5, some real experiments are presented using Khepera III robots and Turtlebot robots. The thesis ends up with major findings and future directions in chapter 6.

CHAPTER 2

LITERATURE REVIEW

2.1 Previous Studies

In the following section, we present some deployment techniques that are classified as in Figure 2.1, then at the end we present some applications of robots deployment.

A geometry based solution for deploying mobile sensors is proposed in [4]. The area is divided into hexagons then each node is instructed by the base-station about the preferred places at the center of a hexagon. Each mobile node perform its calculation to move to the new location. D. Zarbos and et al [7] proposed a

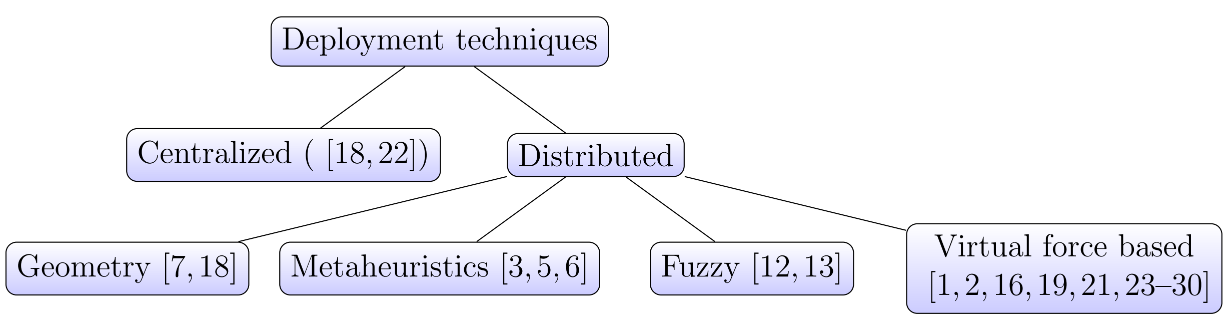


Figure 2.1: Classification of the current deployment techniques

method for redeploying sensors to cover targets. They assume some targets are sparsely covered, so their goal is to balance the coverage of each target in order to enhance network lifetime and connectivity.

In [3], the authors used neural network and genetic algorithms for sensor self-deployment. They found that this approach minimizes the time step and increases the covered area by the learning process and the interaction with the surrounding environment. In [5], the authors proposed a technique for sensor self-deployment in point of interest (PoI). The developed technique requires a priori knowledge of the coordinate of the PoI. It uses the concept of Reduced Neighborhood graph to maintain connectivity during deployment. Another work in [6] is proposed to guarantee near optimal coverage radius around PoI. It uses Greedy Advance and Greedy-Rotation-Greedy. These two algorithms make sensors constructing a triangular tessellation graph around PoI.

In the fuzzy based approach [12, 13], fuzzy logic system is used to control the robot movement. The fuzzy system puts several rules based on the Euclidean distance or the number of robots for example. Then the system will provide a new position that each robot should relocate to it.

In [18], a centralized approach for sensor deployment is presented. The position of demanding nodes and resources is assumed to be known. So the problem is formulated as ambulance allocation problem and an optimization solution is employed.

Virtual force algorithm (VFA) has been used widely to achieve uniform dis-

Ref	Uniform Deployment	Purposeful Deployment	Obstacles	Calibration w_a, w_r	Energy	Notes
[1]	Yes	No	Repulsive	Basic	No	First to use VF
[2]	Yes	No	No	No	No	Require Base station help
[16, 21]	Yes	No	No	No	No	Require high computations
[19]	Yes	No	No	No	No	considers heterogeneous R_s
[24, 25]	No	Implicit	No	No	No	does not handle stuck robots R_s
[26, 27]	Yes	No	No	No	No	requires network wide broadcast R_s
[31]	Yes	No	Repulsive	No	No	considers boundary and different R_s
Proposed work	Yes	Yes	Repulsive and rotate around	Yes	Yes	Handle stuck robots and real implementation

Table 2.1: Virtual force-based deployment techniques

tribution of robots. Table 2.1 provides a summary for virtual force variants that have been used for sensor deployment. The idea of using potential field for robot deployment was first investigated in [17]. In the robotics field, each robot behaves as a source of force for all other robots. This force can be either a positive or negative. If two robots are placed too close to each other, the closeness being measured by a pre-determined threshold, they exert negative forces on each other. This ensures that the robots are not overly clustered, leading to poor coverage in the other parts of the robotics network. On the other hand, if a pair of robots is too far apart from each, they exert positive forces on each other.

In [1], the idea of virtual force was, for the first time, used to improve the coverage after a random deployment of mobile sensors. They consider a binary detection model in which a target is detected (not detected) with complete certainty by the sensor if a target is inside (outside) its circle. After the initial random deployment, all sensor nodes are able to communicate with the cluster head. The cluster head is responsible for executing the virtual force algorithm and managing the one-time movement of sensors to the desired locations. The obstacles are considered in this work and modeled as a repulsive force. This work considers a uniform distribution of the mobile sensor. It is centralized in term of virtual force calculation which is a single point of failure. It did not consider the heterogeneity of the mobile sensors energy level.

G. Tan and et al [2] developed connectivity-preserved virtual force technique such that the covered area is maximized and the connectivity is guaranteed. The developed technique considers that there is a base-station located near the area of interest and the disconnected nodes move toward it to get connected.

Wand et al. [16] added particle swarm optimization (PSO) to virtual force approach. In the process of self-organized deployment, the nodes do not really move, but cluster-head node calculates the virtual path first, and then guides cluster-in nodes to migrate once to save energy. The fitness function of PSO is designed to consider the time cost by self-organized deployment and the coverage rate after deployment. Only a uniform distribution of the mobile sensors is considered in this work and no guidelines on how to choose the virtual force parameters.

Chen et al. [19] considered a probabilistic detection model. In reality, sensor detections are imprecise; hence, the coverage of a target point by a sensor needs to be expressed in probabilistic terms. This work considers the difference of heterogeneous sensor nodes sensing detection radius. They propose a diversity degree to compute distance threshold between nodes in heterogeneous network.

In order to limit the number of neighboring robots involved in virtual force computation, authors in [21] suggest the use of Delaunay triangulation to do so. Robots will only get affected by the attractive force and repulsive force of the nodes that are directly connected to it in the constructed Delaunay triangulation. This approach requires a lot of computation: each node is required to build Delaunay diagram for every iteration of virtual force computations.

Garetto et al. [23] proposed a distributed sensor relocation scheme based on virtual forces, adding the restriction that there are at most only six nodes that can exert forces on the current node. This work handled the problem that arises when nodes are having a high communication range by using the six nodes restrictions. In our work, we handled such problem during the formulation of the virtual force parameters.

Ying et al. use virtual force for post-deployment to improve the coverage of wireless sensor network. They assume that static sensor nodes are initially deployed in the monitoring environment randomly, and the nodes communicate with each other to detect the coverage holes [24]. the mobile nodes will be used for increasing the coverage. Assuming that coverage holes generate an attractive field on mobile nodes, the mobile nodes compute the virtual force for many rounds until there is no force toward the mobile node or the maximum number of rounds is reached so the mobile nodes stay where it stops at the last round. If a force is exerted toward a mobile robot from multiple directions it will cause the robots to oscillate and trigger a lot of unnecessary movements. The same is performed in [25] with particle swarm optimization to reposition the robots to best cover a sensing hole.

The work in [26] aims to overcome the problem of zigzag movement of sensors.

Initially, sensors compute the total virtual force and move based on it; however, in some cases, the sensor may move in a zigzag manner before it reaches the final destination. So, in this work, they propose a prediction mechanism where sensor can predict its position after multiple steps and therefore moves directly to the final position. Instead of moving iteratively, sensors calculate their target position based on an iterative algorithm, move logically, and exchange new logical locations with their new logical neighbors. Actual movement only occurs when sensors determine their final locations. In this approach, all sensors need to be in range of each other to be able to exchange their logical position and compute the next supposed position or require a network wide broadcast of messages which will cause broadcast storm problem.

In [27], they propose a prediction approach for virtual force implementation. Each robot supposed to predict its position after two or more steps in order to avoid zigzag movement and speed up the deployment process. They limit error in [26] by making it only two steps prediction. The main problems with [26] and [27] are that: first, network will behave poorly in case of network partitions. Each partition will predict the position of its robots without a knowledge of positions of the others; second, this mechanism requires a network-wide broadcast which causes a high communication overhead.

Work in [28] adds constraint on when a robot is allowed to move. It considers only one robot to move at a time. It assumes that in the case where all robots relocate themselves at the same moment, some robots may move needlessly. This is because none of the robots makes their movement decision based on the most update robots location information in the neighborhood. So, to solve this problem, they suggest using randomized back off delay to allow only one robot to move at a time. This approach requires synchronization using global communication to determine the required time for each iteration of the algorithm.

Work in [29] considers both obstacles and preferential areas. The obstacles exert a repulsive force based on a rank given to each obstacle, while the preferential areas and target points exert an attractive force based also on a rank given to each preferential point. This works depend on a cluster head to perform all related calculations needed for robot deployment.

Work in [31] considers the boundary effect on virtual force calculations. Sensors are limited in their movement to the boundary of the region of interest. It also limits the number of robots that can participate in virtual force calculation by introducing communication threshold. Robots that are far from each other and beyond the communication threshold will not have an effect on each other, although there are actually communicating.

In [30], the authors use virtual force to perform zone of interest (ZoI) coverage. They compare it with particle swarm optimization and found that virtual force is better in term of coverage but consume more energy.

Robots are used as a guide for human in places such as new shopping malls [32–35] that they are not familiar with. Also, robots can be used for rescue and victim search after damage in buildings [36–38]. The work of [36] introduced a system for tracking robots that are used for discovering victims in an outdoor emergency. The robot is assumed to provide some information regarding the discovery of victims to a base station. This information will be meaningful if it includes the location of the detected victim. So, they used wireless sensor networks to help in localizing the robots using RSSI-based techniques and improved with electronic compass and motor encoders to obtain high accuracy.

Meanwhile, the approach of [37] promotes the establishment of a connection between injured people, who are unable to move, and a base-station using robots. Each civilian is assumed to have a mobile device with communication range M_c , and robots have communication range R_c. The problem is formulated as a mixed integer linear program such that the number of connected civilians is maximized using multi-hop to the base-station.

In [38], the authors assume that there are some places with a potential for some events to happen and mobile robots will be dispatched to help in the places where there is an event. The robots will search for events in a way that can reduce the total energy consumption and balance the load among the robots. The positions of the places for potential events are known and if the number of robots is greater than the number of event places, the problem is formulated as a maximum-matching problem in a weighted bipartite graph. Otherwise, a clustering scheme to group event locations so that the maximum-matching approach can still be applied.

In [32], a WSN is assumed to be deployed before any event may occur. Then it is used to guide people by finding a safe path to the exit. By safest path they mean the path with the largest clearance of the danger zones. Any sensor detects hazard is considered as an obstacle and the safe path is selected such that it is not passing through any obstacles. Another work in [33] uses sensors to help in finding safe evacuation path. Each sensor reports the conditions in its surrounding area to a neighboring decision node (DN). DNs are assumed to make connected network and responsible for computing the safest path. The evacuees are assumed to have smart devices that can communicate with DNs. However, DNs may fail, so in [34] they propose an opportunistic communication by using a mobile decision node (MDN) that are carried by emergency personal or evacuees. MDNs will be localized using static sensors pre-deployed in the building. In case of no safe path and people are trapped, the mechanism proposed in [35] dispatches robots to create escape path by eliminating obstacles from the path.

2.2 Conclusion

As we have seen in all of the work in the literature, virtual force has been used in many applications. However, providing a universal setting for its parameters, specifically the attractive force and repulsive force is missing or not considered carefully. Moreover, most of the work overlooked the purposeful deployment using virtual force and they considered only a uniform distribution of nodes. More importantly, most of these previous works did not consider a real implementation for such deployment. So our aim is to fill these gaps and try to provide a formulation for virtual force factors to meet the demand of different applications.

CHAPTER 3

ROBOTS DEPLOYMENT USING VIRTUAL FORCE APPROACH: GUIDELINES

3.1 Introduction

Robotics network is one of the emerging technologies that has a wide range of applications. Robots are used for many military and civilian applications. Many such applications, such as search-and-rescue operations or area monitoring during an environmental disaster, cannot be effectively carried out by a single robot, but rather are carried out by many robots linked in a robotic network. In rescue operations, for example, robots can be used to help in discovering bodies under the rubbles or even assist the injured. One of the main challenges in these applications is how to deploy the robots without central coordination. Virtual force (VF) technique appears as one of the prominent approaches to perform multi-robot deployment autonomously. It has been proposed in [1] to be used for mobile sensor self-deployment and its calculation is performed as in equation 3.1. However, the setting of its parameters, specifically the attractive force w_a and the repulsive force w_r have not been discussed nor mentioned accurately; they just mentioned that the attractive force should be highly smaller than the repulsive force. So, the effectiveness of this approach depends on how these parameters are selected in order to achieve the required deployment. In this chapter, we will investigate the best setting of these two factors in order to accommodate different kinds of scenarios. The goal is to formulate these two factors to be a function of system parameters such as the number of robots and communication threshold. Additionally, an energy aware implementation of virtual force approach is presented. We consider a scenario where robots have different level of energy and the goal is to balance energy consumption among robots to maximize the lifetime of the network.

Extensive simulation experiments are conducted to study and explore the effectiveness of the proposed settings. In addition, a proof of concept experiment using $Lego^{TM}$ Mindstorm robots is carried out to demonstrate some practical results.

$$F_{ij} = \begin{cases} w_a(d_{ij} - d_{th}), \theta_{ij} & \text{if } d_{ij} > d_{th} \\ 0 & \text{if } d_{ij} = d_{th} \\ \frac{w_r}{d_{ij}}, \pi + \theta_{ij} & \text{if } d_{ij} < d_{th} \end{cases} \tag{3.1}$$

3.2 Methodology

In order to effectively achieve the required deployment, we investigate how to set the attractive force and repulsive force for a different number of robots with different parameters. In addition, the proposed virtual force introduces how to limit the step size that a robot can move in each round to avoid oscillation. Then we introduce an energy aware implementation of virtual force. We consider a scenario where robots have different level of energy and the goal is to balance energy consumption among robots to maximize the lifetime of the network.

3.2.1 Attractive Force (w_a)

In virtual force computation, if robots are far from each other by a distance threshold, which is determined specific to system requirements, the robots will attract each other. Attracting other robots requires the value of w_a to be set properly. If the value of w_a is high, it will cause the robots to move a longer distance than needed and become very close to each other, which will trigger the repulsive force and then attractive force and so on as in Figure. 3.1. In (A) robots are far from each other and the distance between each pair is higher than the distance threshold. In (B) robots became close to each other and the distance between each pair is smaller than the distance threshold. So in (A) robots will keep attracting each other to become as in B, then robots will again repulse each other and return to the state in (A) and so on. Consequently, the robots will not stabilize and the network convergence would be difficult to attain.

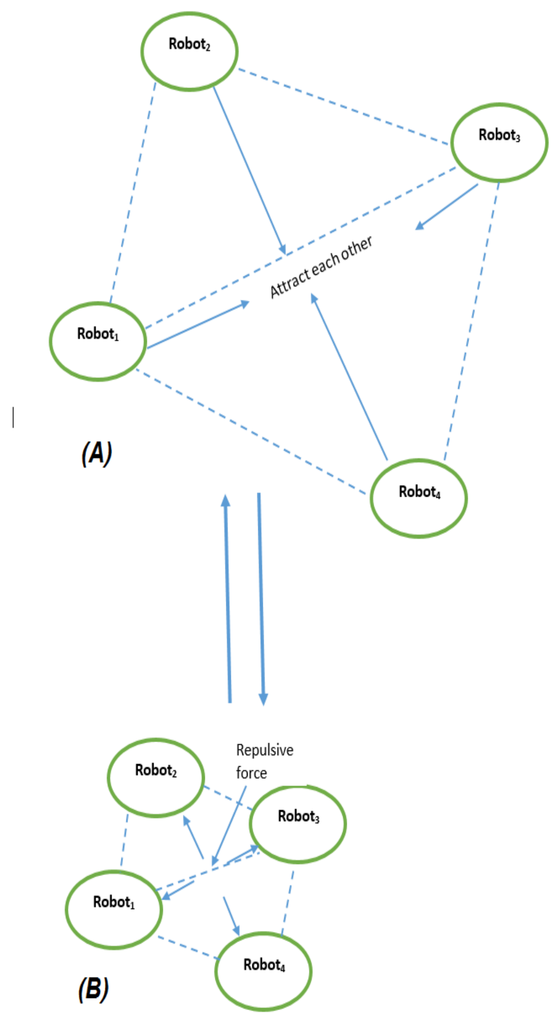


Figure 3.1: An example of robots oscillation when there is no control over the attractive force. A) The initial positions of the robots. B) The new positions of the robots after applying the attractive force in A

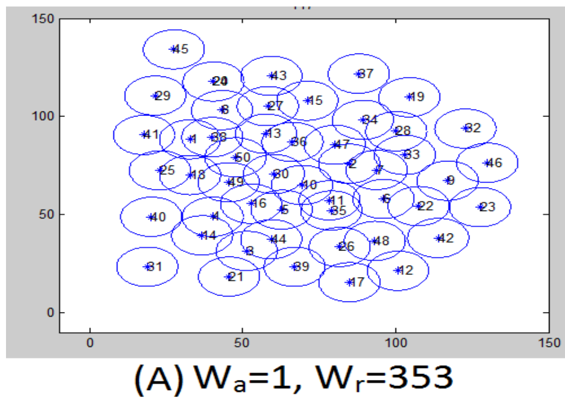


Figure 3.2: The effect of a high number of robots on the attractive force setting

If the number of robots is high, the attractive force should be reduced to avoid that a robot attracts a high number of robots to its vicinity $F_a \propto \frac{1}{N}$. For example, in Figure. 3.2, when the number of robots is high (N=50), setting the attractive force to be high (i.e. $w_a = 1$) caused the robots to remain close to each other, although their repulsive force was set properly.

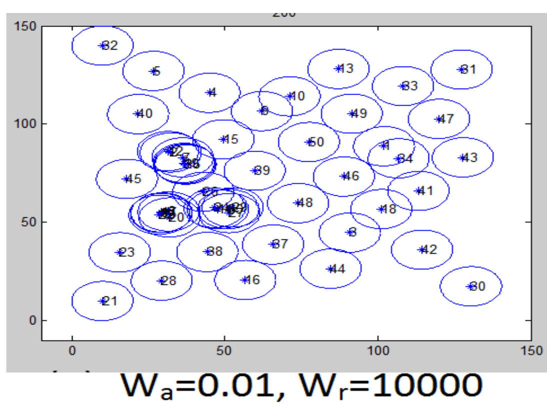


Figure 3.3: An example of w_a set to a random small value

Additionally, setting the attractive force arbitrarily to be smaller compared to the repulsive force did not give the expected result as shown in Figure. 3.3.

Furthermore, if the communication range is high, this means that the effect of one robot will be on a higher number of robots. In this case, it is better to set

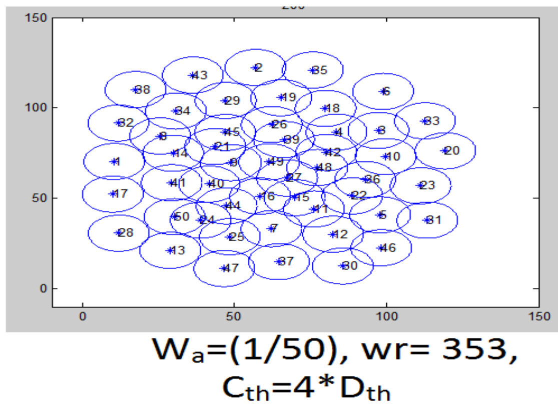


Figure 3.4: An example to show the effect of high communication range on virtual force based deployment

the attractive force to decrease with the increase in the communication threshold $F_a \propto \frac{1}{R_c}$. In Figure. 3.4, when the communication range is high (i.e., $R_c > 4R_s$), it yields nonuniform distribution of the robots.

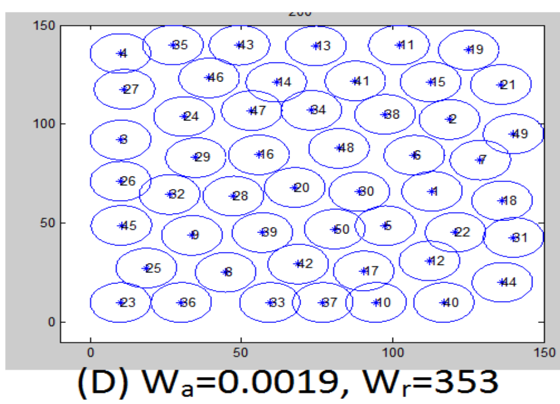
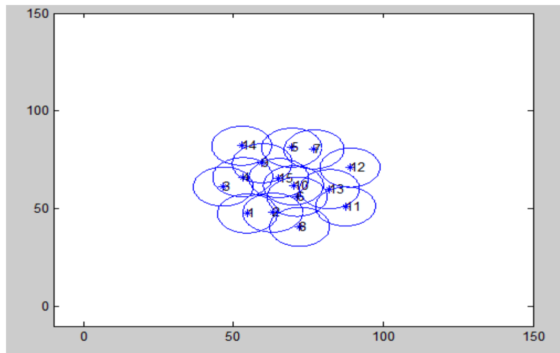


Figure 3.5: An example of the achieved deployment using the proposed setting of the attractive force

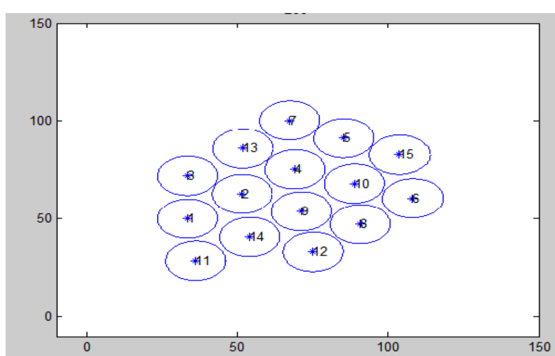
So, for a high number of robots, the best setting is shown in Figure. 3.5, where the attractive force considers the number of robots as well as their communication range according to the formula discussed below.

On the other hand, if the number of robots is small, the better to have the attractive force high to keep the robots close to each other. In Figure. 3.7, we



(A): $W_a=1$, $W_r=100$

Figure 3.6: An example of setting w_a for a small number of robots



(B): $W_a=0.2$, $W_r=58$

Figure 3.7: An example of setting w_a for a small number of robots

show different settings for w_a under a small number of robots, we see that setting w_a to be high is suitable but to a certain level, otherwise, as in Figure. 3.6, it causes the robots to attract each other more and not spreading uniformly.

So, in order to accommodate all the situations aforementioned, we propose the following setting for the attractive force:

$$w_a = \frac{D_{th}}{R_c} * (\text{number of robots})^{-\alpha} \tag{3.2}$$

Where D_{th} is the distance that should be kept between each two robots at the final deployment which is $D_{th} = M * R_s$ and $M < 2$, R_c is the communication range of each robot which is $R_c = K * R_s$, and $K > 2$, α is an arbitrary but predetermined tuning parameter, e.g., α can take the value 2 (in the presented experiments α =3/2), and the number of robots represents a number of mobile robots. Increasing the value of α increases the repulsive force and decreases the attractive force. Decreasing the value of α decreases the repulsive force and increases the attractive force.

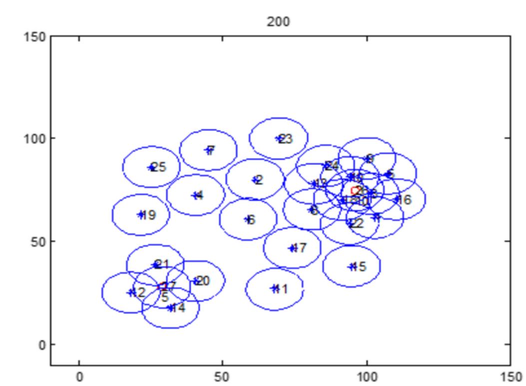


Figure 3.8: The attractive force for preferential points (point 26 and 27)

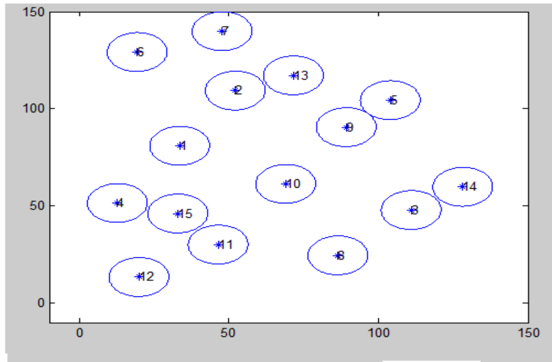
For a preferential point, the location that should attract more robots, the setting of W_a^i should be higher relative to other robots setting of w_a , and proportional to the preferential need to be given to that point. In Figure. 3.8, we see that points 26 and 27 have different preferential, and the preferential of point 26 is higher than point 27. As a result, the point 26 was able to attract more robots to its vicinity. The more attractive force is given to a point by multiplying its w_a by a factor relative to the attraction of the others. For example, in Figure. 3.8, point 26 and 27 require a demand of 10 and 5 robots respectively. So, point 26 has w_a multiplied by 10, however, point 27 has w_a multiplied by 5.

3.2.2 Repulsive Force w_r

Having discussed the attractive force key role in virtual force approach, however this approach still depends on another factor which is the repulsive force (w_r). The repulsive force is triggered when the robots are close to each other by less than the distance threshold. The setting of this parameter has great influence on the stability of the virtual force approach.

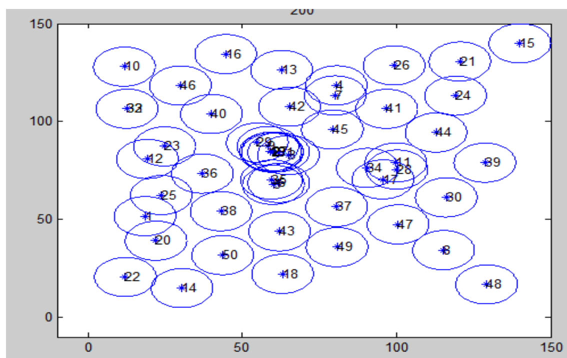
If this parameter is set to a high value, it will have a different impact if the number of robots is high or low. If the number of robots is low and w_r is high relative to their number, it will cause them to become out of range of each other as shown in Figure. 3.9. This kind of behavior is solved partially by limiting the step size that a robot can move at one time.

On the other hand, if the number of robots is high, and the repulsive force



(C): W_a=0.02, W_r=500

Figure 3.9: The effect of the repulsive force w_r when the number of robots is low. N=15, w_r=500



(A): W_a=0.0019, W_r=3000

Figure 3.10: The effect of the repulsive force w_r when the number of robots is high N=50, w_r=3000

is high beyond a certain level ($i.e \geq 1000$), it will cause the robots to move in clusters as shown in Figure. 3.10 and Figure. 3.11. So controlling the value of w_r has the same impact of w_a . As a result, we propose that w_r is to be proportional to the number of robots based on the following equation:

$$w_r = (\text{number of robots})^{\alpha} \tag{3.3}$$

As the number of robots increases, the repulsive force should increase to let the robots to spread throughout the area and maximize the coverage.

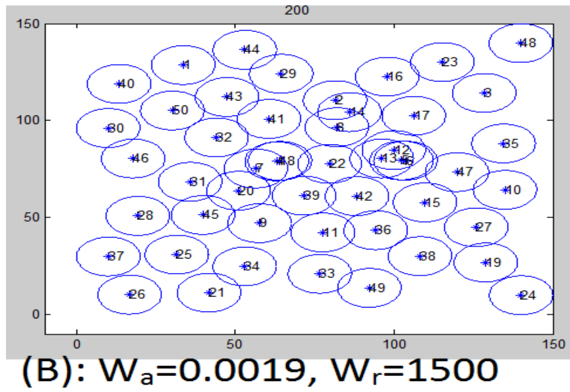
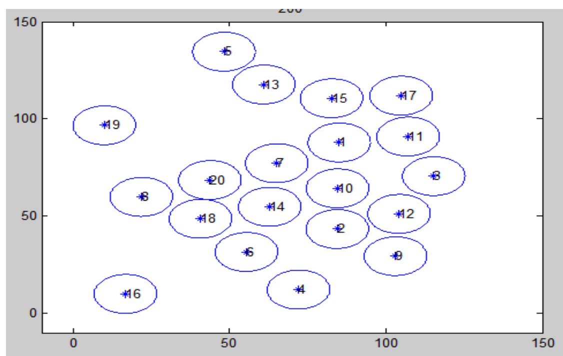


Figure 3.11: The effect of the repulsive force w_r when the number of robots is high N=50, w_r=1500



Without limit step size

Figure 3.12: The effect of step size being not controlled

In all the above settings, the step size should be limited, otherwise the robots will not stabilize at all. For example, if the robots are very close to each other, the repulsive force will be high and could cause them to move higher distance that would result in robots getting out of range of each other. On other cases, it could trigger high attractive force, then high repulsive force, then again high attractive force and so on, so the network will not stabilize at all. This effect is shown in Figure. 5, where the number of robots is 20. The step size could be any value less than the R_s, however, if its value is too small, it will delay the convergence and trigger a high number of communication messages. Limiting step

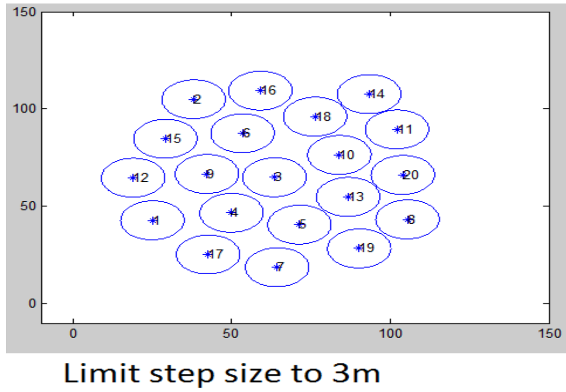


Figure 3.13: The effect of controlling the step size

size produces the required uniform distribution of the robots as well as reasonable way of movements.

3.2.3 Energy Aware Virtual Force (EAVF)

In order to consider the level of energy that each robot has, we include the remaining energy in virtual force calculation.

So Virtual force computation will be according to the following equation:

$$F_{ij} = \begin{cases} w_a(d_{ij} - d_{th}) * E_{factor}, \theta_{ij} & \text{if } x > d_{th} \\ 0 & \text{if } d_{ij} = d_{th} \\ \frac{w_r}{d_{ij}} * E_{factor}, \pi + \theta_{ij} & \text{if } x < d_{th} \end{cases} \tag{3.4}$$

$$E_{factor} = 1 + \frac{E(i) - E(j)}{max(E(i), E(j))} \tag{3.5}$$

So the energy of each pair of robots is considered such that if the robots are with the same energy level, then the original VF will work, otherwise the one with the higher energy will be under higher VF from those with lower energy and vice versa. Additionally, since we consider that there is a limit on the step size that a robot can move, this limit will be related to the level of energy of the current robot and its neighbor as a summation of the difference between robots energy and its neighbors energy as follows:

$$N_e = \sum_{k=1}^{N} \frac{E(i) - E(k)}{N} \tag{3.6}$$

1. If $E(i) \leq 10\%$ of the maximum energy, the robot moves a very small step and keep participating in virtual force messages.

2. If $N_e \leq -10\%$of the max_e, then the step size is decreased by a certain level.

3. If $N_e \geq 10\%$ of the max_e, then the step size is increased by a certain level. Otherwise the step level stays as the default.

3.3 Results and Discussion

We have tested the above formulas of w_a and w_r on different scenarios with different requirements. Moreover, we have tested virtual force approach with different settings of w_a to study its relationship with different number of robots. As we can see from the results in Figure. 3.14 and Figure. 3.15, for different number of robots, there is a safe region where the value of w_a provides the highest performance in term of coverage maximization. If we would like to give some robots

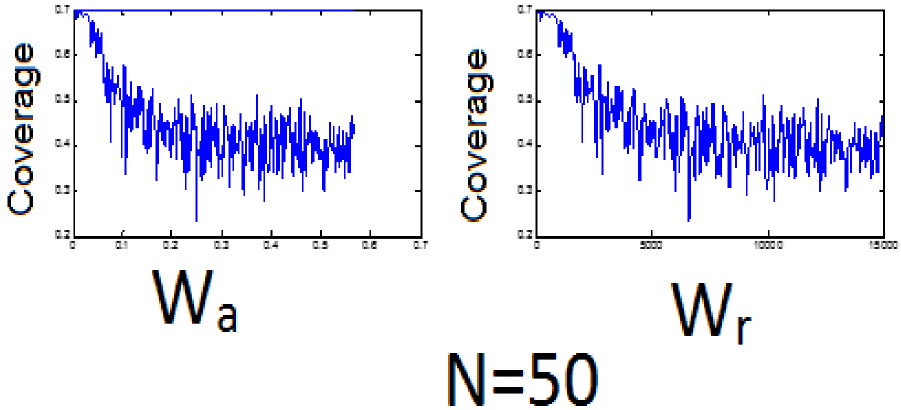


Figure 3.14: Coverage level, with number of nodes=50, Dth=20, and area size= 150*150m

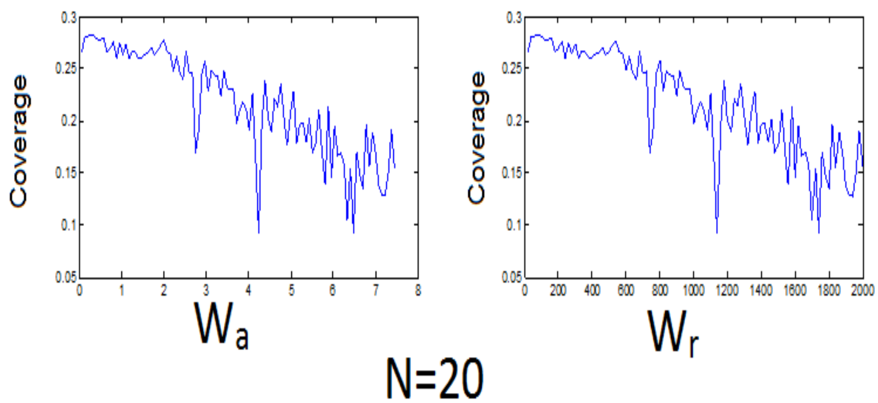


Figure 3.15: Coverage level, with number of nodes=20, Dth=20, and area size= 150m*150m

preferential over the others, the other robots should have the same w_a value while the w_a of the preferred robots should be set higher relative to how much preference we would like to put. The proposed formula for w_r is also tested under different variations and different scenarios. We see in Figure. 3.14 and Figure. 3.15 that when a small number of robots are used, we have a limit on how much the attractive force could be set. On the other hand, increasing the number of robots imply increasing the repulsive force but to a certain level, after that the performance of the network in term of coverage deteriorates. The proposed formulas for w_a and w_r fall on the safe region for any number of robots, which provides an insight of how the proposed formulas can adapt to different kind of scenarios.

To show the effect of considering energy, we have conducted some experiments using the original virtual force and the energy aware virtual force. We found out that the proposed approach is able to balance the energy consumption among the robots with high remaining energy and those with low level of remaining energy as in Figure. 3.16. In the basic virtual force, all the robots, regardless of their remaining energy, almost has traveled the same distance, while with energy aware version, the robots with low energy level (EAVF-low) have traveled less distance compared to those with high energy level (EAVF-high).

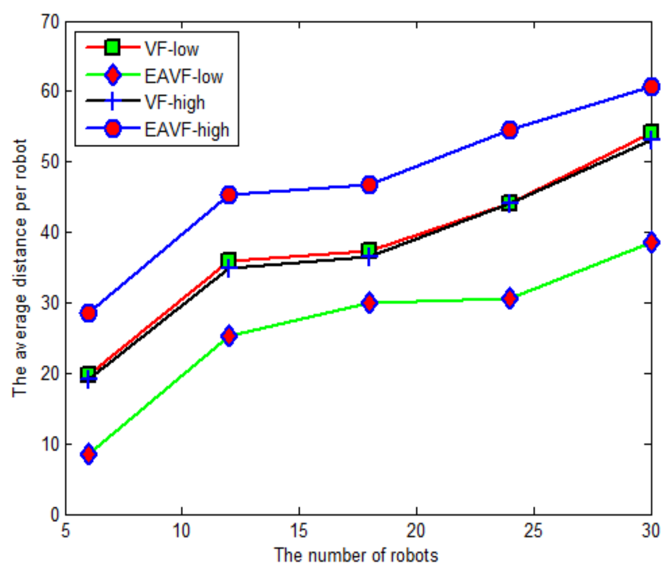


Figure 3.16: The distance traveld by robots with low and high level of energy using VF and EAVF

Additionally, we see that applying the energy-aware version does not affect much the coverage level achieved of robots deployment. In both approaches, the basic and the energy-aware one, almost the same level of coverage has been achieved as in Figure. 3.17.

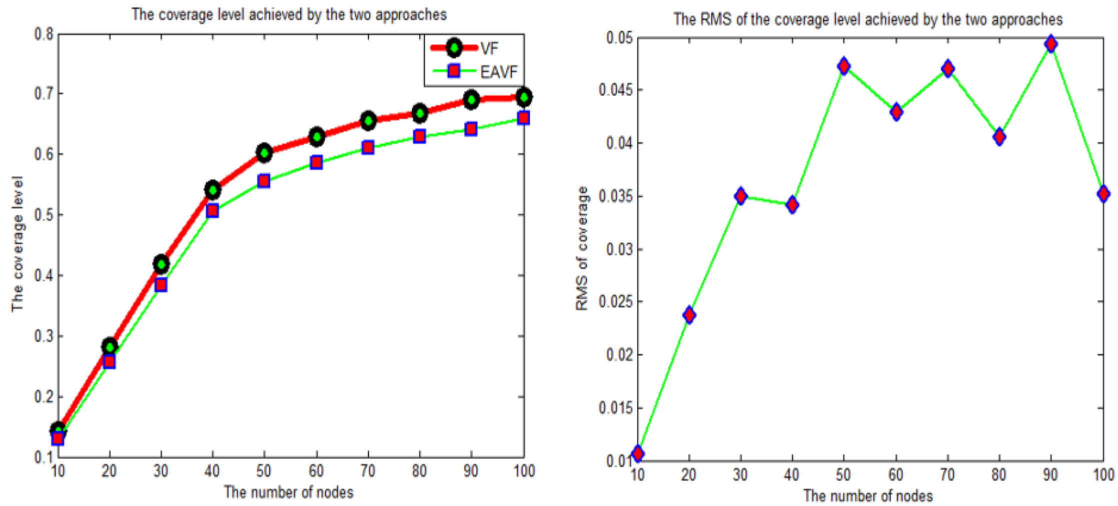


Figure 3.17: A) The coverage level achieved using the energy-aware version and the original version. B) The root mean square of the difference between the coverage level of the two approaches.

We have also studied the sensitivity of choosing the neighboring energy threshold (N_e) to see its effect. We tested values range from 0.03 to 1. For each value the number of robots is from 6 to 30, half of them with low energy and the second half with high energy. Then we measure the sensitivity by computing the root mean square (RMS) of the difference between the average traveled distance by low energy robots and high energy robots. Figure. 3.18 shows that increasing the level of the threshold gives a good balance in the distance traveled by robots with low and high energy which is expressed as RMS until it reaches around point 0.3 where the RMS starts to decline. That gives us an insight of how we can choose the threshold value to attain the best result.

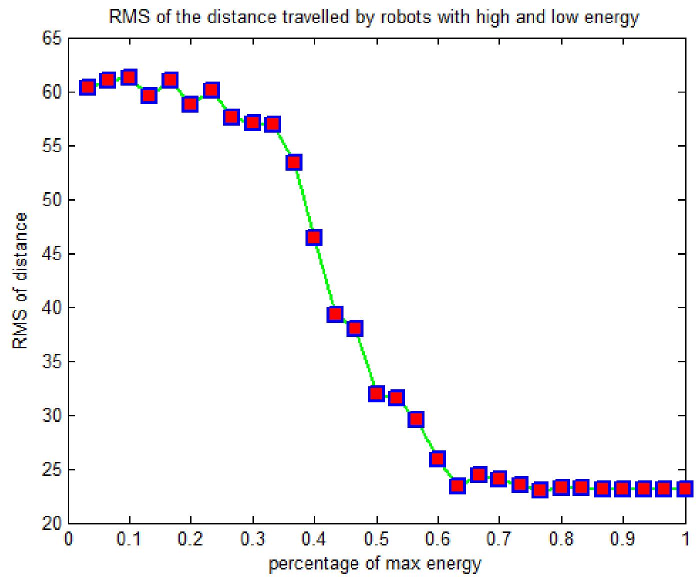


Figure 3.18: The sensitivity of choosing N_e threshold

3.4 Virtual Force Testbed Using EV3 Robots

In order to test the proposed setting of w_a and w_r , we have carried out a few experiments using a testbed consisting of six Lego Mindstorm EV3 robot. The Education EV3 Core Set (45544) consists of EV3 programmable brick, two Large Motors, one Medium Motor, two Touch Sensors, one Color Sensor, one Gyroscopic Sensor, one Ultrasonic Sensor, cables, USB cable, one Rechargeable battery and many technical elements. We installed tiny Linux distribution (leJOS) on a microSD in order to implement our algorithm in each robot and be able to program it using java. The experiments have been conducted on two scenarios: one scenario where the robots are close to each other and the repulsive force is expected to work in this case, and the second scenario where the robots are far from each other and the attractive force is supposed to be triggered for such scenario. Figure. 3.19 shows snapshots for the first scenario while Figure. 3.20 shows snapshots for the second scenario. Using the proposed setting of w_a and w_r , the robots were able to achieve the required deployment in both Figure. 3.19 and 3.20.

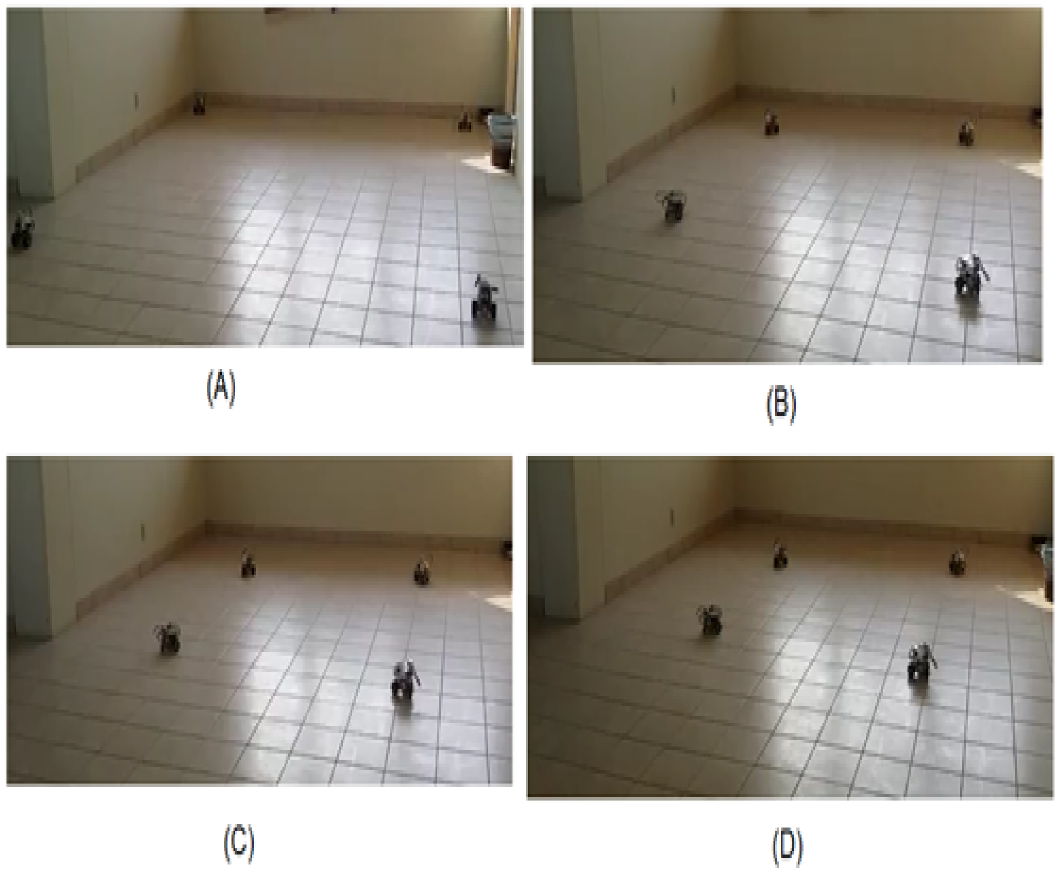


Figure 3.19: Attractive force test using 4 EV3 robots. A) The initial position of the robots. B) Robots after moving one step. C) Robots after moving another step. D) Final deployment.

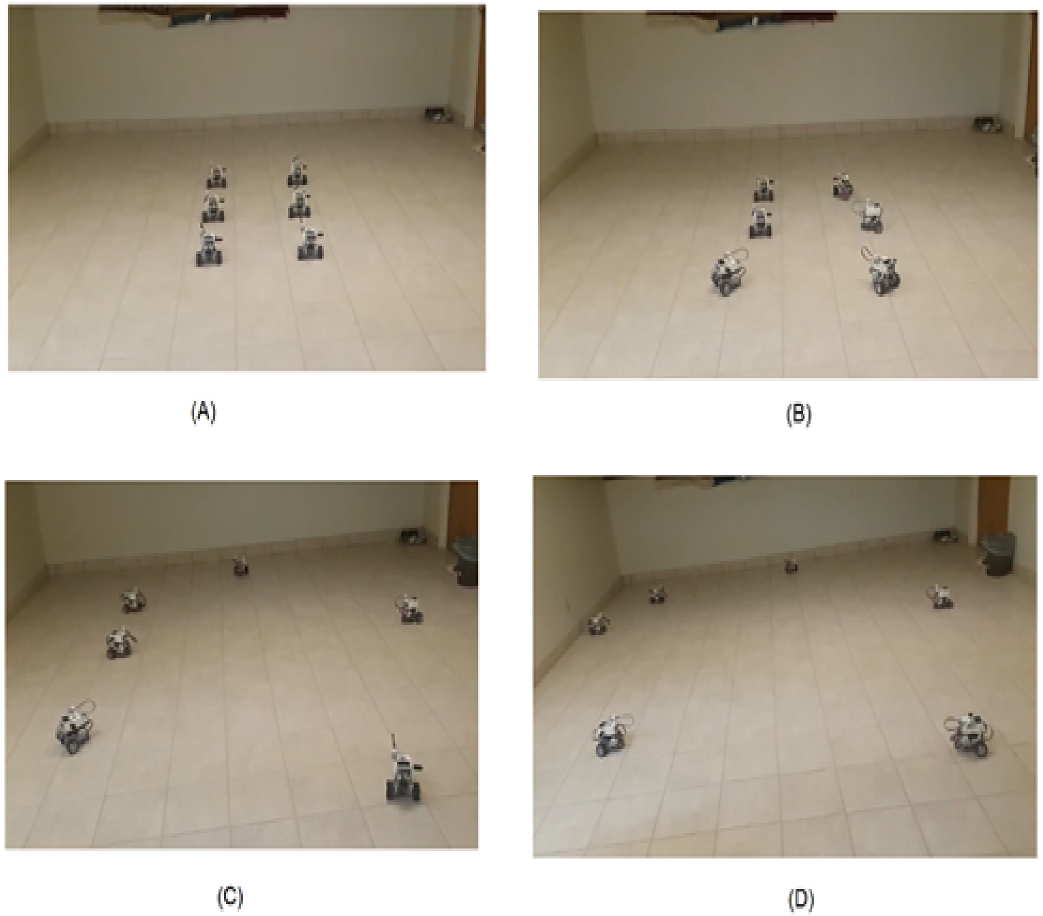


Figure 3.20: Repulsive force test using 6 EV3 robots. A) The initial position of the robots. B) Robots after moving one step. C) Robots after moving another step. D) Final deployment.

3.5 Conclusion

In this chapter, we have investigated how to set virtual force parameters in order to accommodate different scenarios. We propose two calibrates one for the attractive force and one for the repulsive force which factor system parameters such as the number of robots while computing attractive/repulsive forces. The proposed settings have been tested using simulation and in practice using Lego EV3. The results show that the proposed settings reach the ultimate coverage level, adapt to the number of nodes, the area size and the preferential points demand. Moreover, we have presented an energy-aware version of virtual force that is able to balance the remaining energy among robots and at the same time achieve the required deployment.

CHAPTER 4

A COOPERATIVE VIRTUAL FORCE FOR ROBOT DEPLOYMENT

4.1 Introduction

Employing a networked set of robots is an effective way to serve applications in areas where human intervention is impossible or possess risks. In rescue operations, for example, robots can be used to help in discovering bodies under rubbles or even assist the injured. Collaboration among robots will be essential in these applications in order to efficiently achieve the allotted goals in a timely manner. Realizing such collaborative operation without a central coordination is a key challenge. Some works have proposed methods for the distribution of robots, but these have tended to suffer from limitations such as evenly spreading the robots regardless of demand, requiring an a priori known distribution of the demand over an area, or requiring centralized coordination of the robots. This work tackles these challenges in scenarios where landmarks are present in the deployment area. The robot deployment problem is modeled as an optimized node coverage based on the individual landmarks. We formulate such dynamic coverage problem using Potential Fields where landmarks and mobile robots exert virtual forces based on the landmarks' demand and the mutual distance between them. In this chapter, we present a Cooperative Virtual Force Robot Deployment (COVER) technique. As mentioned earlier, virtual force (VF) technique appears as one of the prominent approaches to perform multi-robot deployment autonomously. However, most of the existing VF based approaches lack purposeful deployment. Our approach modifies the original VF approach to overcome this problem and considers the mission requirements such as the number of required robots in each locality (e.g., landmarks are distributed and each needs a specific number of robots in its vicinity). In addition, COVER expedites the deployment process by establishing a cooperative relation between robots and neighboring landmarks. We then found out that COVER suffers from limitations in some scenarios such as deadlock or increase in the total distance moved by each robot. So, in order to overcome such limitations, we also propose a modification of COVER called Two-hop COVER to address the aforementioned problems. It improves the performance of COVER by utilizing two-hop communication to speed-up the process of satisfying landmarks' demand. Lastly, Two-hop COVER may fail in some scenarios to reach

100% demand satisfaction so we propose a Trace Fingerprint method that should guarantee 100% demand satisfaction when Two-hop COVER fails to do so.

4.2 Problem Statement and System-level Assumptions

We consider an area of interest A that has a set of landmarks (N). The landmarks are used to guide the robots deployment process. A set of the landmarks N' is equipped with special capabilities, e.g., sensing and computational resources, to enable them to assess the situation in their vicinity and request the presence of a number of robots (d) to perform certain tasks. A set of robots R is initially randomly deployed in (A). The goal is to develop a distributed mechanism for the robots self-deployment such that the requirements of each landmark are met. The following enumerates the key system model assumptions:

1. Each landmarks node knows its location.

2. Robots are homogeneous, i.e., have the same speed, service capabilities, energy supply, etc.

3. Each landmark can request a number of robots depending on the need of services in its area.

4. Landmarks can communicate with each other and exchange information.

5. Each robot knows its initial position.

6. The positions of landmarks and their demand are unknown to the robots.

4.2.1 System Model

Denote R to be a set of robots initially dropped at any point in the area of interest. N is the total number of robots and let i denote each specific robot, where $i = 1, ..., N$. Each robot has a communication range R_c, within its range it can communicate with other robots and landmarks. Denote L to be a set of landmarks distributed randomly in the area of interest. The number of landmarks is M and lets j denotes each landmark where $j = 1, ..., M$. Each landmark has a demand $d(j) \geq 0$, the demand is represented by a number of robots that should be around a given landmark. Any robot can be in one of two states: free or associated. Free robots are those which are not associated with any landmark yet. Denote R_f to be the set of free robots, initially $R_f = R$. Associated robots are those that successfully got associated to a landmark (R_i, L_m). Denote R_a to be the set of associated robots. The aim is to make the number of associated robots $|R_a|$ equal to the total demand of the landmarks (i.e $|R_a| = \sum\{d(j)\}$). Let's denote the landmark that a robot associated with by L_{R_i}. Each robot R_i has a set of neighbor robots $N_r(R_i) = \{R_n : |R_{x_i,y_i} - R_{x_n,y_n}| < R_c, n \neq i, n = 1, ..., N\}$ and neighbor landmarks $N_l(R_i) = \{L_m : |R_{x_i,y_i} - L_{x_m,y_m}| < R_c, m = 1, ..., M,)\}$. The neighbor robots of robot R_i can be either free $N_r(R_i) \in R_f$ or associated $N_r(R_i) \in R_a$. And the neighbor landmarks can be either satisfied (i.e. $d(N_{L_j}(R_i)) = 0$) or not satisfied (i.e. $d(N_{L_j}(R_i)) > 0$).

4.3 COVER: a COoperative Virtual FoRcE Deployment

4.3.1 Procedure

R Robots will be initially dropped at any point in the area of interest. Robots will utilize virtual force among themselves to spread over the area and to improve the chances of locating landmarks that are having demand. Each robot computes the composite virtual force and moves accordingly. In each move, each robot stops for a while to collect messages from other robots and landmarks to decide its next step. Any robot may receive two kinds of messages.

1. Messages from other robots $R_{fRep} \subseteq R_f$ that are not yet associated with any landmarks. These messages are treated normally as in the basic virtual force (i.e. the robots will utilize them to compute either attractive force or repulsive force depending on the distance to the source robot as in equation 4.2).

2. Messages from landmarks $L_{replies} \subseteq L$ and/or other robots $R_{aRep} \subseteq R_a$ that are already associated with a landmark. These messages are explicitly stating the kind of force that should be utilized by the receiver of the message. For each robot $R_i \in R_{aRep}$, if $d(L_{R_i}) > 0$ or $d(L_j \in N_l(R_i)) > 0$, then the robot R_i will exert an attractive force. The attractive force is toward its landmark (L_{R_i}) or one of the neighboring landmarks (the one with the highest demand). Otherwise, it will exert a repulsive force to increase the

chance that a robot will move in a direction where it can find landmarks with demand. For each landmark $L_j \in L_{replies}$, if the demand $d(L_j) > 0$, the messages will contain a demand request. Otherwise, if $d(L_j \in N_l(L_j)) > 0$, then the landmark L_j will exert an attractive force toward one of the neighboring landmarks (the one with the highest demand). Otherwise, it will exert a repulsive force.

The landmarks announce their demands in terms of a specific number of robots. Robot R_i that hears the demand message will add that landmark to a list (demand list) $dl = \{(L_k, d(k))|k = 1, ..., N_l, where N_l = |N_l(R_i)|\}$ in order to respond to the nearest one based on the Euclidean distance. In order to avoid more than needed robots move toward one landmark, an association process is proposed as shown in Figure 4.1. Each robot first sends an association message to the nearest landmark $L_k \in dl$. Then, if $d(L_k) > 0$, it will reply with a confirmation message. Otherwise, it will send a rejection message. If the robot receives a confirmation message, it will move toward that landmark to a position determined by the landmark itself. If the robot receives a rejection message, it will contact the next landmark in its dl if it has already heard from multiple landmarks. If the robot failed to associate to any landmark, it will proceed by computing the composite virtual force as shown below and move accordingly.

The virtual force calculation will be modified to consider the aforementioned

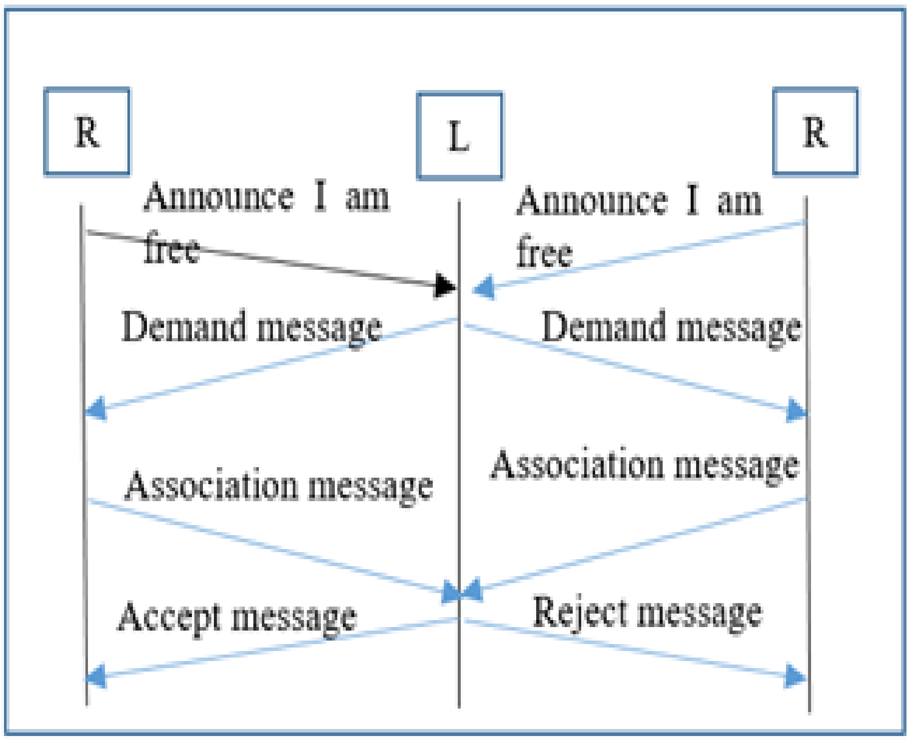


Figure 4.1: Timeline diagram to show the association process, R denotes a robot and L denotes a landmark

cooperation. The total force will be as in equation 4.1.

$$F_i = \sum_{j=1, j \neq i}^{K} F_{ij} + \sum_{a=1, a \neq i}^{M1} F_{ia} + \sum_{r=1, r \neq i}^{M2} F_{ir} \tag{4.1}$$

$j \in N_{R_f}(R_i))$, and $M1$ is the robots $\in N_{R_a}(R_i)$ and $d(N_l(R_a)) > 0$ and the landmarks $\in N_{L_s}(R_i)$ and $d(N_l(L_s)) > 0$ where L_s is the set of satisfied landmarks, however, $M2$ is the robots $\in N_{R_a}(R_i)$ and $d(N_l(R_a)) = 0$ and the landmarks $\in N_{L_s}(R_i)$ and $d(N_l(L_s)) = 0$.

In computing the composite virtual force, we differentiate between three cases. The first case when the neighboring robots are not associated with any landmark, then, the calculation goes as the basic virtual force [1] based on 4.2.

$$F_{ij} = \begin{cases} w_a(d_{ij} - d_{th}), \theta_{ij} & \text{if } x > d_{th} \\ 0 & \text{if } d_{ij} = d_{th} \\ \frac{w_r}{d_{ij}}, \pi + \theta_{ij} & \text{if } x < d_{th} \end{cases} \tag{4.2}$$

where

$$w_a = \frac{d_{th}}{C_{th}} * (\text{number of robot})^{-\alpha} \tag{4.3}$$

and

$$w_r = (\text{number of robot})^{\alpha} \tag{4.4}$$

The second case is when the received messages come from associated robots or from landmarks where the demand in their proximity is greater than zero, the calculation is as follows.

$$F_{ia} = w_a * d_{ia} \tag{4.5}$$

where

$$w_a = d(j) * (\text{number of robot})^{-\alpha} \tag{4.6}$$

α is an arbitrary but predetermined tuning parameter, e.g., α can take the value 2 (in the presented experiments α =3/2), and the number of robots represents a number of mobile robots. Increasing the value of α increases the repulsive force and decreases the attractive force. Decreasing the value of α decreases the repulsive force and increases the attractive force.

The third case is when the received messages come from associated robots or from landmarks where the demand in their proximity is zero, the calculation goes as follows.

$$F_{ir} = \frac{w_r}{d_{iR}} * \alpha \tag{4.7}$$

The flow of the cooperative virtual force calculation is shown in Figure. 4.2. We see that each received message will pass through the filter presented in the flowchart. First, if the message is a normal message, then based on the Euclidean distance the force will be computed. Otherwise the message is either attractive or repulsive and will be treated according to its type.

The procedure of the COVER approach is presented in algorithm 1, and 2. The details of robots operation are in algorithm 1 and for the landmarks the details are in algorithm 2.

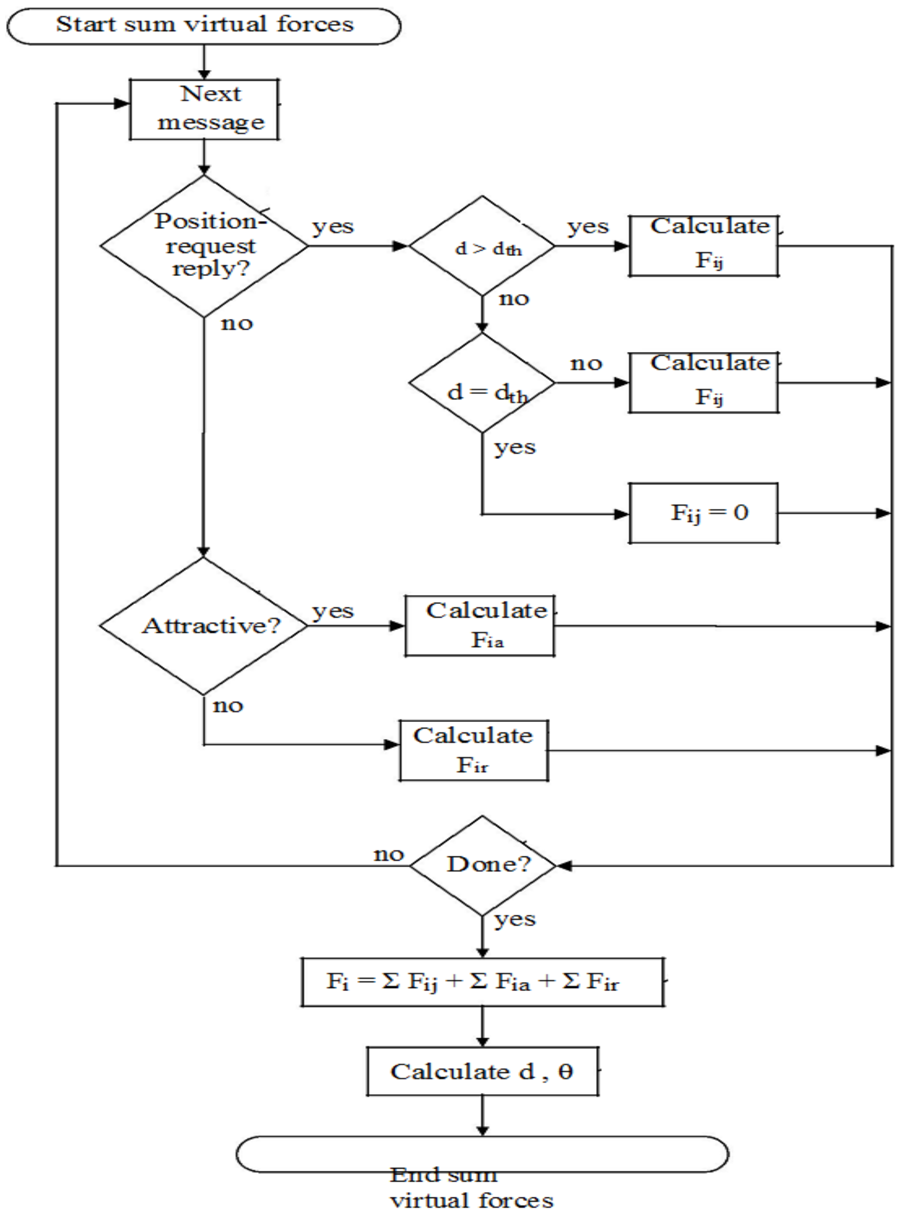


Figure 4.2: Virtual force calculation according to COVER approach

Algorithm 1 The algorithm for the operation of the robot according to COVER

switch Robot_status **do**

  case unassociated

    R_i sends a neighbors position inquiry message

    R_i receive position reply messages from robots ($R_k \in N_r(R_i)$) and landmarks $L_k \in N_l(R_i)$:

    for all replies of $L_k \in N_l(R_i)$ **do**

      if $d(L_k) > 0$ **then**

        Add L_k to the potential demanding landmarks list dl.

      end if

    end for

    function ATTEMPASSOCIATION(dl)

      Repeat

      Choose the nearest landmark in dl and associate to.

      if succeed **then**

        Mark this robot as associated

        Change Robot_status to associated

      else

        Remove this landmark L_k from the list dl

      end if

      Until dl is empty

    end function

    if unassociated **then**

      Compute the composite virtual force (VF) according to equation 4.1

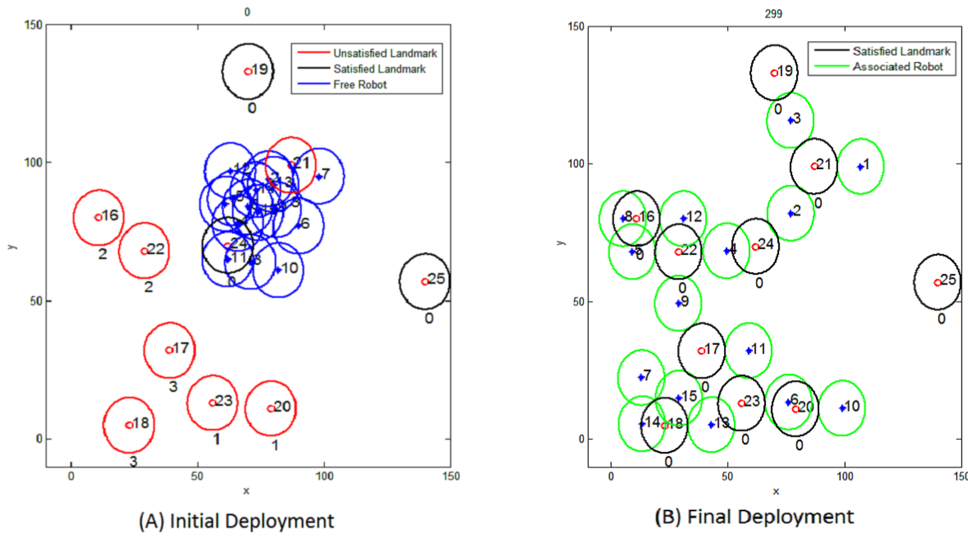
      Compute the new position and relocate to it

    end if

```
case associated
    if $R_i$ receives position request message then
        if $d(L_{R_i}) > 0$ then
            Apply $w_a \propto d(L_{R_i})$
        else if $d(N_l(R_i)) > 0$ then
            Pick the one with the highest demand
            Reply with its position and demand
        else
            work as a repulsive force
        end if
    end if
```

Algorithm 2 The algorithm for the operation of the landmark according to COVER

for all landmarks L **do**
    if L_j receives a position request message from a robot **then**
        if $d(L_j) > 0$ **then**
            Reply with its position and demand.
        else if $d(L_j) == 0$ & $d(N_l(L_j)) > 0$ **then**
            Pick $L_k \in N_l(L_j)$ that has the highest demand.
            Reply with its position and demand.
        else
            work as a repulsive force
        end if
    else if receive association message **then**
        if $d(L_j) > 0$ **then**
            Reply with an accept message, and a position to come to it.
        else
            Reply with a reject message.
        end if
    end if
end for



(a) The initial positions of the robots and the demand of the landmarks.

(b) The final positions of the robots and the remaining demand of the landmarks.

Figure 4.3: An example of the cooperative landmarks and robots using virtual force. The number below the red circle is the demand of that landmark. Black circles are landmark with demand zero.

4.3.2 Detailed Example

In the scenario presented in Figure. 4.3, we have 10 landmarks numbered (16 to 25) with demand vector [2, 3, 3, 0, 1, 3, 2, 1, 0, 0], respectively. We placed 15 robots in the center of the area. They will use virtual force among them in order to spread throughout the area and search for landmarks. Once a robot receives a message from a landmark, it starts responding to it. If it receives from multiple landmarks it will respond to them, one by one based on predefined criteria; here, we consider the distance between robot and landmark as the decisive factor to join a certain landmark. In this scenario, all the landmarks will get their demands. The cooperative approach helped landmark 18 to get its need through robot 9, 11 and landmark 17. First, robot 14 gets closer to robot 9; robot 9 knows that the landmark 18 has a demand and its landmark (landmark17) is satisfied with its demand, so it will attract robot 14 toward the landmark 18. The same with landmark 17 and robot 11 for attracting robot 7 and 13 toward landmark 18.

4.3.3 Simulation Setup

Parameters	**Value**
Simulation tool	Matlab
Number of robots (randomly distributed)	15, 20, 25, 30, 35
Number of landmarks (randomly distributed)	10
Total landmarks demand (randomly distributed)	10, 15, 20, 25, 30, 35
Waiting time	3 sec
Robots transmission range	50 m
Landmarks transmission range	50 m
Area size	150m x 150m
Stopping criterion	Total force 0

Table 4.1: Simulation Parameters for COVER Approach

In order to evaluate the proposed cooperative virtual force, we have conducted extensive simulation experiments examining the effectiveness on different setups. The simulation is implemented using Matlab using the parameters in table 4.1. The performance metrics used in this study are:

1. Demand satisfaction: this metric measures the percentage of demand that is satisfied by the end of the implementation of the algorithm. Thic metric shows how effective is the proposed solution in term of how much percentage

of the demand is satisfied.

2. The total traveled distance: this metric is used to measure the total distance traveled by the robots in order to achieve the level of demand satisfaction.

3. The total time needed to achieve the demand satisfaction reached by each approach. It is the time until the last associated robot reaches the position determined by its landmark. We aim to reduce the time needed to achieve the maximum demand deployment especially for time critical applications.

4. Total messages: this metric counts the number of messages that are exchanged in the implementation of COVER algorithm. The messages are mainly due to virtual force messages and the proposed cooperation messages.

All the above metrics can be used to implicitly measure the energy consumption because the total distance and messages are the main sources of energy consumption.

We have compared COVER with three other approaches namely, Hungarian algorithm [22] (centralized approach), basic virtual force (Basic VF) approach [1], and full virtual force (Full VF) [39] [24]. We describe briefly each approach as follows:

1. Hungarian algorithm (Centralized approach): our problem is that we have a set of resources R (robots), and a set of demand D of landmarks. The Hungarian method solves this problem by assigning the best robots to each

landmark based on the distance between robots and landmarks, its complexity is $O(n^3)$.

2. Basic VF: in this approach, the robots will use the original virtual force to deploy themselves in the area of interest. Once any of the robots hear a demand message from a landmark, it will start the association process. If it succeeds, it will move toward that landmark, if not it will continue moving using virtual force. It is basically designed for uniform deployment not for a purposeful deployment.

3. Full VF: in this approach, the robots will implement the virtual force until they reach the equilibrium state. If a robot receives any demand message from landmarks, it will include the landmarks demand in virtual force calculations. So, in order to compute the virtual force by each robot, the demand messages from landmarks are considered and will be calculated as in equation 4.6. Only the landmarks that are having demand can exert the attractive force, otherwise zero force is exerted. The force is set relatively to the demand needed by the landmark; the higher the demand, the higher the attractive force. After finishing the deployment, the robots will start listening for the demand messages and applying the association algorithm, described above, to get associated with one of the neighboring landmarks.

4.3.4 Result and Analysis

The performance evaluation of COVER solution is considered under different scenarios. In the first scenario, the total demand is set to be equal to the number of available robots. The total demand is distributed randomly between landmarks, so we may have landmarks with the demand of zero and others with the demand greater than or equal one. The initial positions of the robots are set randomly in a compact zone in the middle of the area. Each experiment is repeated until we reach 95% confidence level with all the parameters are randomly generated. The same simulation setup is implemented for the other approaches considered for comparison.

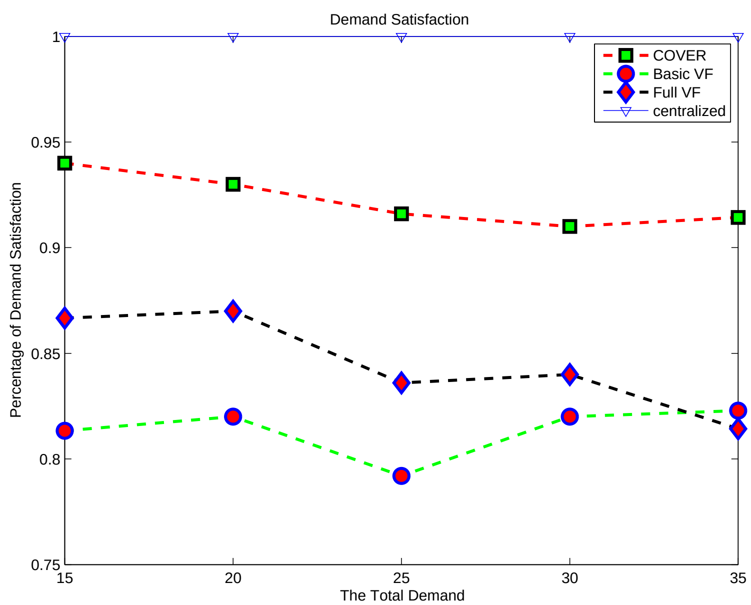


Figure 4.4: The percentage of demand satisfaction, the number of robot= the total demand, area size=150m x 150m, communication range=50m

Figure 4.4 shows the percentage of demand satisfaction achieved using our proposed approach. For instance, we can observe that when the demand changes from 15 to 35, COVER achieves 95% demand satisfaction. The centralized approach achieves 100% demand satisfaction because in the centralized approach the number of landmarks, their demand, and their positions should be known beforehand in contrary to our proposed solution where all these factors are not known. So, in COVER algorithm, there are two tasks; the first is to search for landmarks that have demands, and the second is to satisfy the demand so the load on the robots is doubled. Furthermore, COVER algorithm outperforms all other heuristics. This is due to the cooperative behavior implemented amongst landmarks and robots. We can also see that Full VF achieves a high level of demand satisfaction because this approach has some sort of cooperation between landmarks and robots but it is less than COVER approach. Moreover, it takes longer time to satisfy the demand and longer distance as well as we can see in Figure. 4.5 and 4.6.

Robots	**Centralized**	**COVER**	**%**	**Basic**	**%**	**Full**	**%**
15	641.80	780.00	22%	785.60	22%	1157.20	80%
20	841.90	1001.20	19%	1053.90	25%	1695.10	101%
25	1107.20	1347.40	22%	1471.30	33%	2349.70	112%
30	1333.70	1658.20	24%	1837.80	38%	3148.90	136%
35	1493.60	1927.20	29%	2280.10	53%	4049.50	171%

Table 4.2: The total distance in meter for each approach for each number of robots. The percentage shows the added total distance compared to the centralized approach

We also measure the total distance traveled by the robots to achieve the level demand satisfaction attained by each approach. We can see in Figure. 4.5 that robot in COVER approach travels 20-30% more distance compared to the centralized approach. It is also shown that the total distance in the basic virtual force is

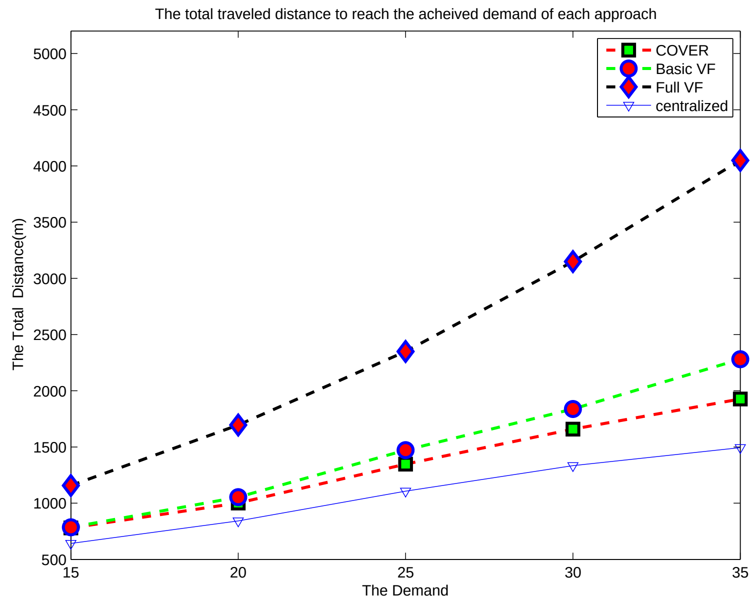


Figure 4.5: The total traveled distance to achieve the level of demand satisfaction in figure 4.4, the number of robots=the total demand, area size=150m x 150m, communication range=50m

the same as that of the COVER approach for small number of robots and higher when the number of robots exceeds 20 as in Table 4.2 although COVER approach was able to achieve a higher level of demand satisfaction as in Figure. 4.4. The higher the level of demand satisfaction, the more movements of robots will be caused which is the case for COVER. This is due to the guidance incorporated by the COVER approach. In addition, we can notice that COVER is by far better than the Full VF in term of the distance traveled. The Full VF travel around 100% more distance compared to COVER approach and around 150% compared to the centralized as in Table 4.2.

Deployment duration is an essential factor to assess any algorithm. Therefore, we have recorded the total time needed for robots in order to achieve the attained level of demand satisfaction by each approach. The total time is shown in Figure.

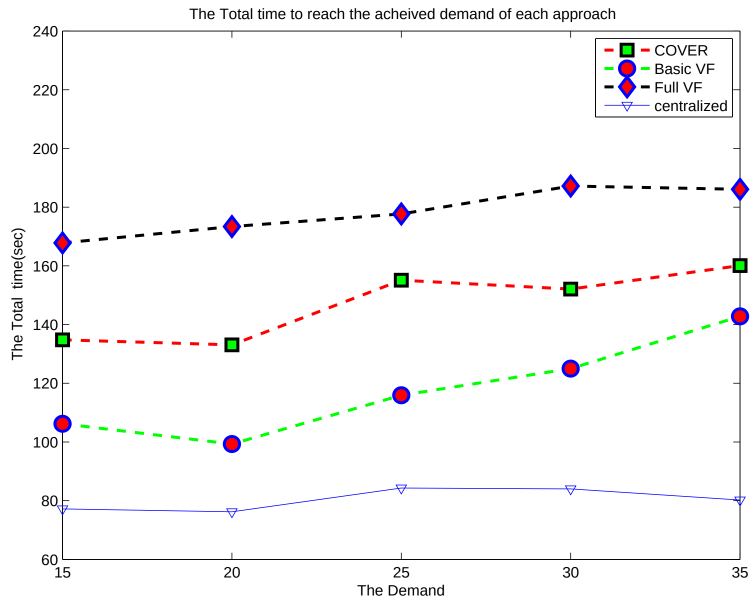


Figure 4.6: The total time needed to achieve the level of demand satisfaction in figure 4.4, the number of robots=the total demand, area size=150m x 150m, communication range=50m

Robots	**Centralized**	**COVER**	**%**	**Basic**	**%**	**Full**	**%**
15	77.20	134.80	75%	106.20	38%	167.80	117%
20	76.20	133.10	75%	99.30	30%	173.40	128%
25	84.30	155.10	84%	115.90	37%	177.70	111%
30	84.00	152.10	81%	125.00	49%	187.20	123%
35	80.20	160.10	100%	142.80	78%	186.10	132%

Table 4.3: The total time in seconds for each approach for each number of robots. The percentage shows the added total time compared to the centralized approach

4.6. We observe that COVER takes around 10-20% more time compared to the Basic VF, this is due to the high level of demand satisfaction achieved by COVER. The higher the achieved level of demand satisfaction, the more time will be used by robots to relocate to a position determined by the landmarks. We report the total time needed by each approach to reach the level of demand satisfaction achieved in Figure. 4.4. Since COVER achieves a high level, it takes longer time. This includes the time until the robot reaches the position determined by each landmark as shown in Table 4.3.

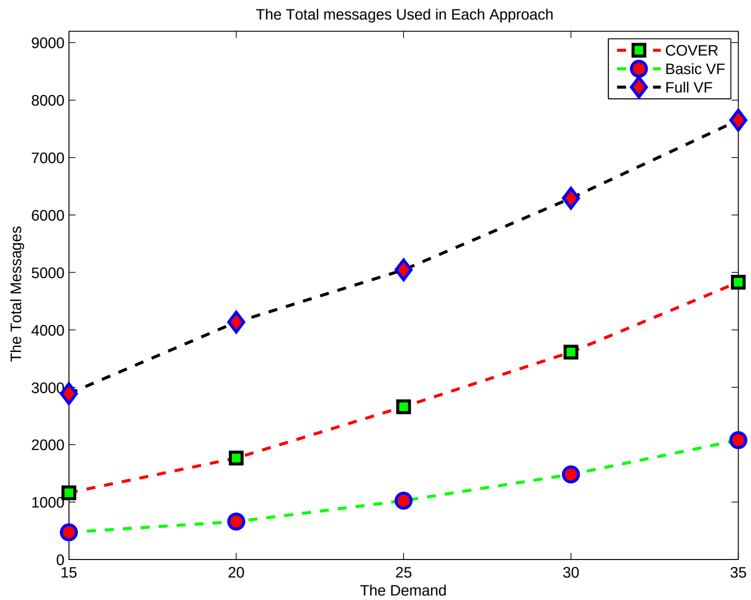


Figure 4.7: The total number of messages used in each approach, the number of robots= the total demand, area size=150m x 150m, communication range=50m

Another important factor is the total number of messages exchanged during the implementation of each approach. We can observe in Figure 4.7 and Figure 4.11 that the number of messages used in COVER is higher than the basic virtual force, which happens for two reasons. First, COVER technique relies on the cooperation between robots and landmarks, which necessitates more messages. Second, COVER technique achieves higher demand satisfaction which means more messages to be used for the association purpose. Compared with Full VF, COVER needs less number of messages which demonstrates the effectiveness of the employed cooperative approach in achieving the highest demand satisfaction with minimum messages overhead.

In the second scenario we would like to see the effect of changing the demand while the number of robots is constant. We put the number of robots 25, while the demand of the landmarks varies from 10 to 30. The other settings are the same as the first scenario. We see that the increase in the demand for the same number of robots in COVER technique yields a high percentage of demand satisfaction compared to the other approaches and its performance is very close to the centralized approach when the demand is less than 20. Although when the demand is 10, all the approaches yield the same performance in term of the demand satisfaction as in Figure. 4.8, they perform differently when it comes to the number of exchanged messages and the total travelled distance as shown in Figure. 4.9 and Figure. 4.10. In Figure. 4.9, the traveled distance to achieve the level of demand satisfaction for the COVER technique is less than the basic virtual force by around 15-25%. This is again because COVER technique is designed to enhance the demand satisfaction. For the time needed to reach the achieved demand, we see that COVER takes longer time by around 20% to reach a higher level of demand satisfaction when the demand is greater than 20 because COVER achieved a higher level of demand satisfaction compared to Basic VF as shown in Figure. 4.10.

We have also studied the effect of using different transmission ranges. As we have seen, all the above scenarios were with a transmission range of 50 meters. So, we will show the effect of increasing the transmission ranges from 50 meter to 90 meter. First, in Figure. 4.12 we see that increasing the transmission range helps in improving the demand satisfaction until it almost reaches 100% demand satisfaction when the transmission range is 90. The higher the transmission range,

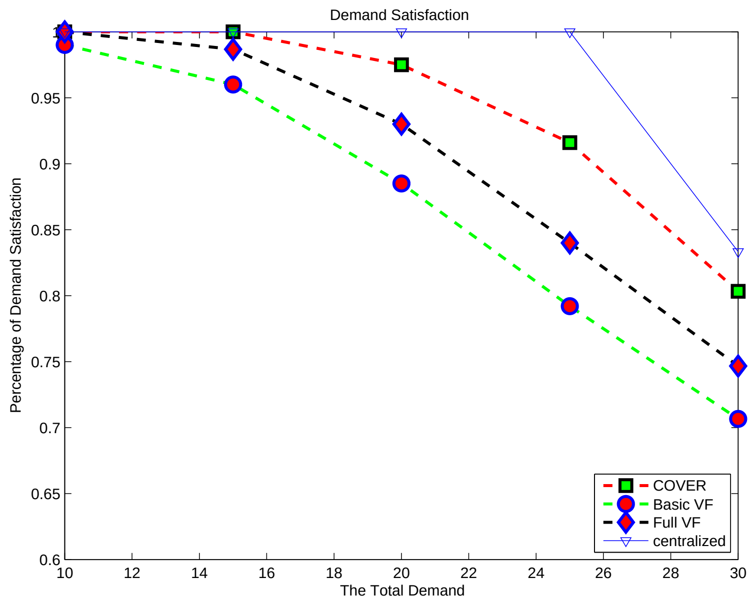


Figure 4.8: The percentage of demand satisfaction, the number of robot= 25, area size=150m x 150m, communication range=50m

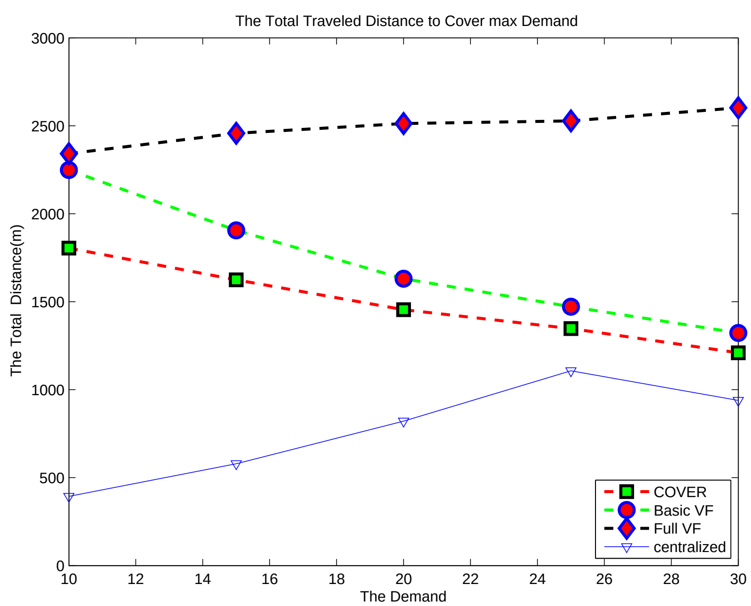


Figure 4.9: The total travel distance, the number of robots=25, area size=150m x 150m, communication range=50m

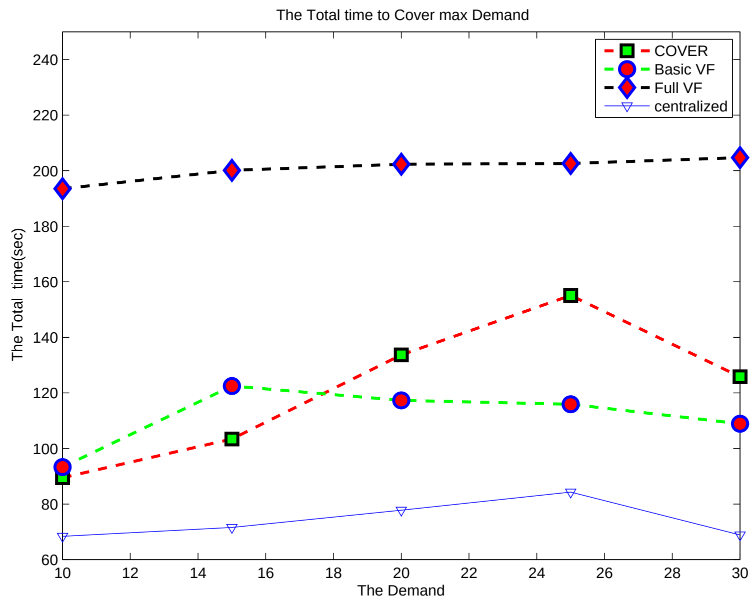


Figure 4.10: The total time, the number of robots=25, area size=150m x 150m, communication range=50m

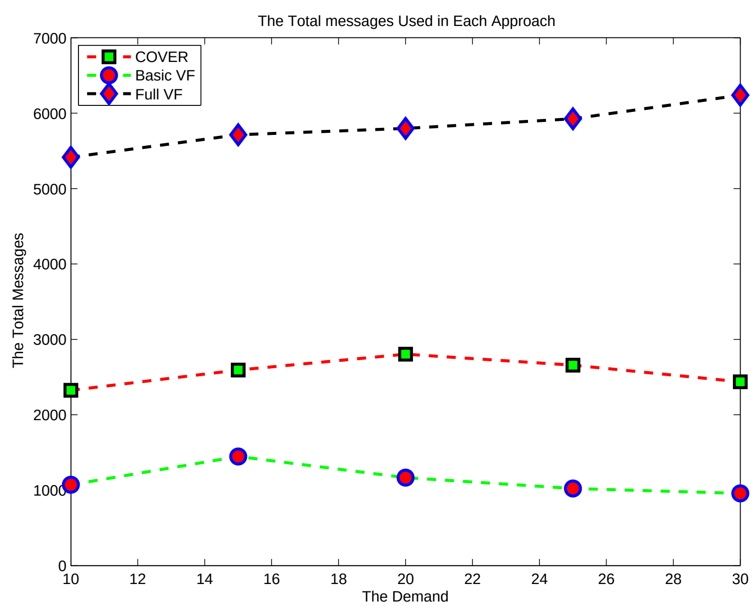


Figure 4.11: The total messages, the number of robots=25, area size=150m x 150m, communication range=50m

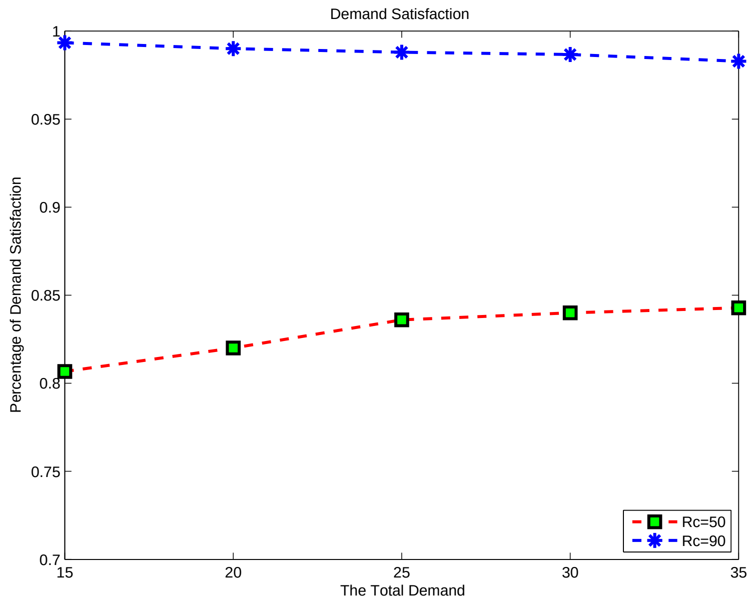


Figure 4.12: The percentage of demand satisfaction for COVER algorithm, the number of robots=the total demand, area size = 200m x 200m and different transmission range

the easier it becomes for a robot to locate a landmark and associate to it. Also, the cooperation scale will increase and each associated robot or satisfied landmark will be able to help other unsatisfied landmarks in their new higher transmission range.

Moreover, the distance per associated robot will decrease with the increase in the transmission range. But, since we have a high number of associated robots, it means that each associated robot will move an additional distance to relocate to a position determined by its landmark so the total distance remains the same when we increase the transmission range as we see in Figure. 4.13.

For the total time needed to achieve the required deployment, the increase in the transmission range to 90m reduces the total time by around 20% compared to 50m as in Figure. 4.14 although a higher level of demand satisfaction acheived

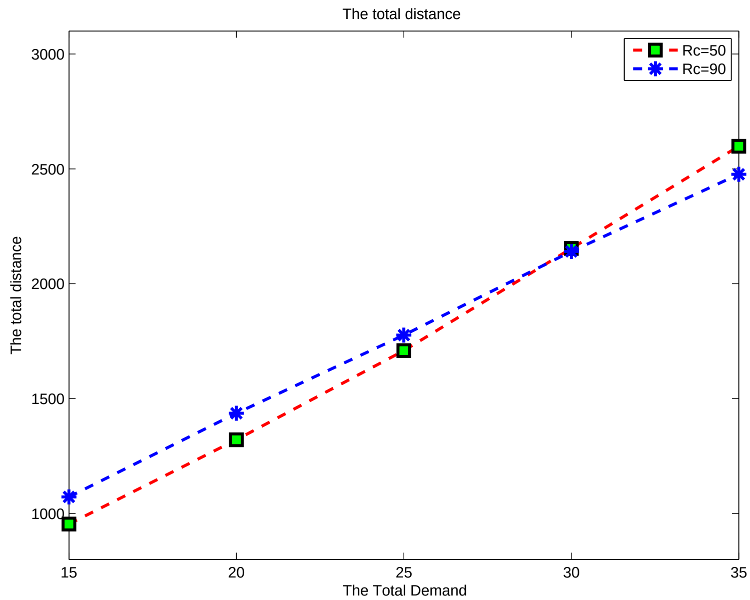


Figure 4.13: The total distance traveled by all robots to reach the achieved level of demand satisfaction for COVER algorithm, the number of robots=the total demand, area size = 200m x 200m and different transmission range

for 90m communication range means that additional time will be needed by some robots to relocate to a position determined by their landmarks.

Moreover, the increase in the transmission range will help the robots to get associated so quickly to landmarks and consequently reduces the total messages by around 10% as well as in Figure. 4.15.

4.3.5 Algorithm Analysis

As we have seen above, COVER algorithm was successfully able to reach a high level of demand satisfaction. In this section we will analyze its performance and see how effective it is.

Here we will analyze each of the following to find out how approximately each metric can be calculated.

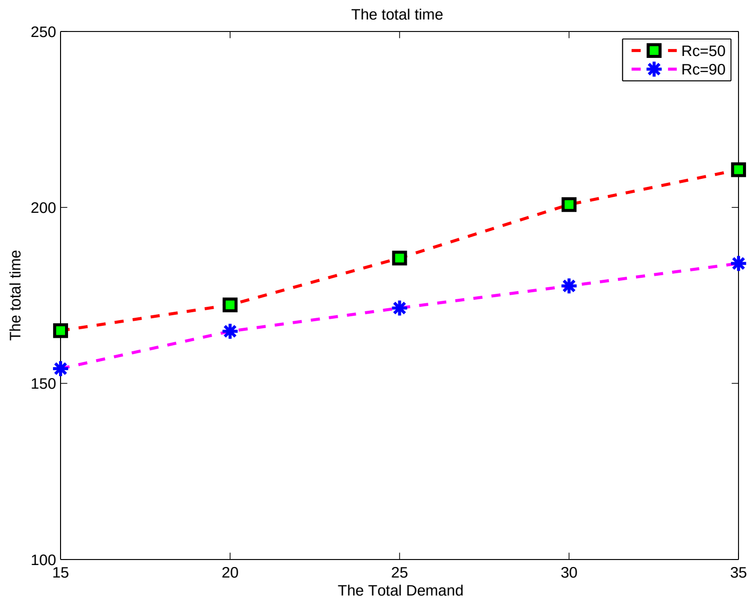


Figure 4.14: The total time needed to reach the achieved level of demand satisfaction for COVER algorithm, the number of robots=the total demand, area size = 200m x 200m and different transmission range

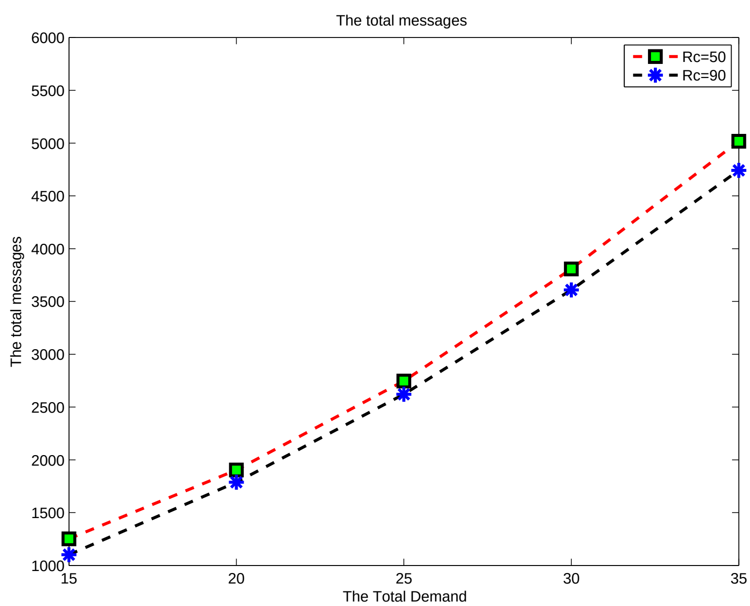


Figure 4.15: The total messages exchanged till the end of implementing the COVER, the number of robots=the total demand area size = 200m x 200m and different transmission range

1. Total messages: the total number of messages each robot will utilize during COVER implementation will be as follow.

   (a) The broadcast message at the beginning of each time slot until the robot gets associated.

   (b) The replies from neighbor robots and landmarks.

   (c) The cooperation messages sent and received by the neighboring satisfied landmarks and associated robots.

   (d) The association messages.

Lemma 4.1 *The total number of messages a free robot will utilize until it gets associated is:*

$$M_{total}(R_i) = \sum_{t=1}^{T} M_b + M_r(t) + M_c(t) + M_a(t) \tag{4.8}$$

where T is the number of slots before the robot gets associated. M_b is the number of broadcast messages, M_r is the reply messages, M_c is the cooperation messages, and M_a is the association messages.

Proof.

In each time slot t, the total number of broadcast messages will be

$$M_b(t) = |R_f| \tag{4.9}$$

Then for each broadcast message, the replies will be from all robots and landmarks in the range of the sender.

$$M_r = |N_r(R_{M_b})| + |N_l(R_{M_b})| \tag{4.10}$$

So, the total reply messages in a given slot is

$$M_r(t) = \sum_{j=1}^{k} |N_r(R_j)| + |N_l(R_j)| \tag{4.11}$$

where k is the number of free robots at time slot t.

When an associated robot $\in R_a$ or a satisfied landmark $\in L_s$ receives a broadcast position request message, it will inquire its neighbor landmarks to see if they have demand. So the total messages to cooperate with R_j will be as follow. The total coopative broadcast messages:

$$M_{cb}(t) = |N_{Ra}(R_j)| + |N_{Ls}(R_j)| \tag{4.12}$$

and the total replies for the cooperation broadcast messages will be

$$M_{cr}(t) = |N_l(R_a)| + |N_l(L_s)| \tag{4.13}$$

where $R_a \in N_{Ra}(R_j)$ and $L_s \in N_{Ls}(R_j)$

So, the total cooperation messages in each time slot will be

$$M_c(t) = M_b(t) * M_{cb}(t) * M_{cr}(t) \tag{4.14}$$

If a robot gets in range of some landmarks that are having demand not satisfied, it will contact them until it gets associated to one of them. If the number of neighbor landmarks $N_l(R_i)$ is n_1, then the robot may need to contact 1 to n_1 landmarks until it gets associated. So, in the slots before the robot gets associated, it will contact all n_1 landmarks, so the total association messages is

$$M_a(t) = n_1 \tag{4.15}$$

∎

2. The total time:

Lemma 4.2 *The total time needed for a robot to get associated is:*

$$Total\_time = \sum_{t=1}^{T}(t_r(t) + t_a(t) + t_v(t)) + t_{landmark\_relocation} \tag{4.16}$$

Proof. In each time slot, each robot will send a position request message and wait for replies t_r. If the robot receives demand messages from landmarks, it will contact them to get associated with one of them, this time is called t_a. If the robot succeeded to associate with one of the landmarks, it will relocate to a position determined by that landmark which is referred to as $t_{landmark\_relocation}$. If it failed to associate, it will compute the distance it should move according to COVER. This distance is less than max_{step_size} and will take t_v. ∎

3. The total distance:

Lemma 4.3 *The total distance traveled by a robot to get associated is:*

$$Total\_distance = \sum_{t=1}^{T}(d_{VF}(t)) + d_{landmark\_relocation} \quad (4.17)$$

Proof. Each robot movements will be either due to virtual force calculations d_{VF} or the distance to relocate to a position determined by its landmark. The d_{VF} will be $\in [0, max_{step\_size}]$. The distance needed to get to a position determined by the associated landmark is $d_{landmark\_relocation}$. ∎

4.4 Twp-hop COVER

4.4.1 Introduction

As we have noticed in the previous section, COVER was unable to fulfill completely the demand of all landmarks. Also, the total time and traveled distance are a little bit high. So, in this section, we will introduce an improved version of COVER called Two-hop COVER with the aim to increase the level of demand satisfaction and reduce the total time and traveled distance. The first amendment

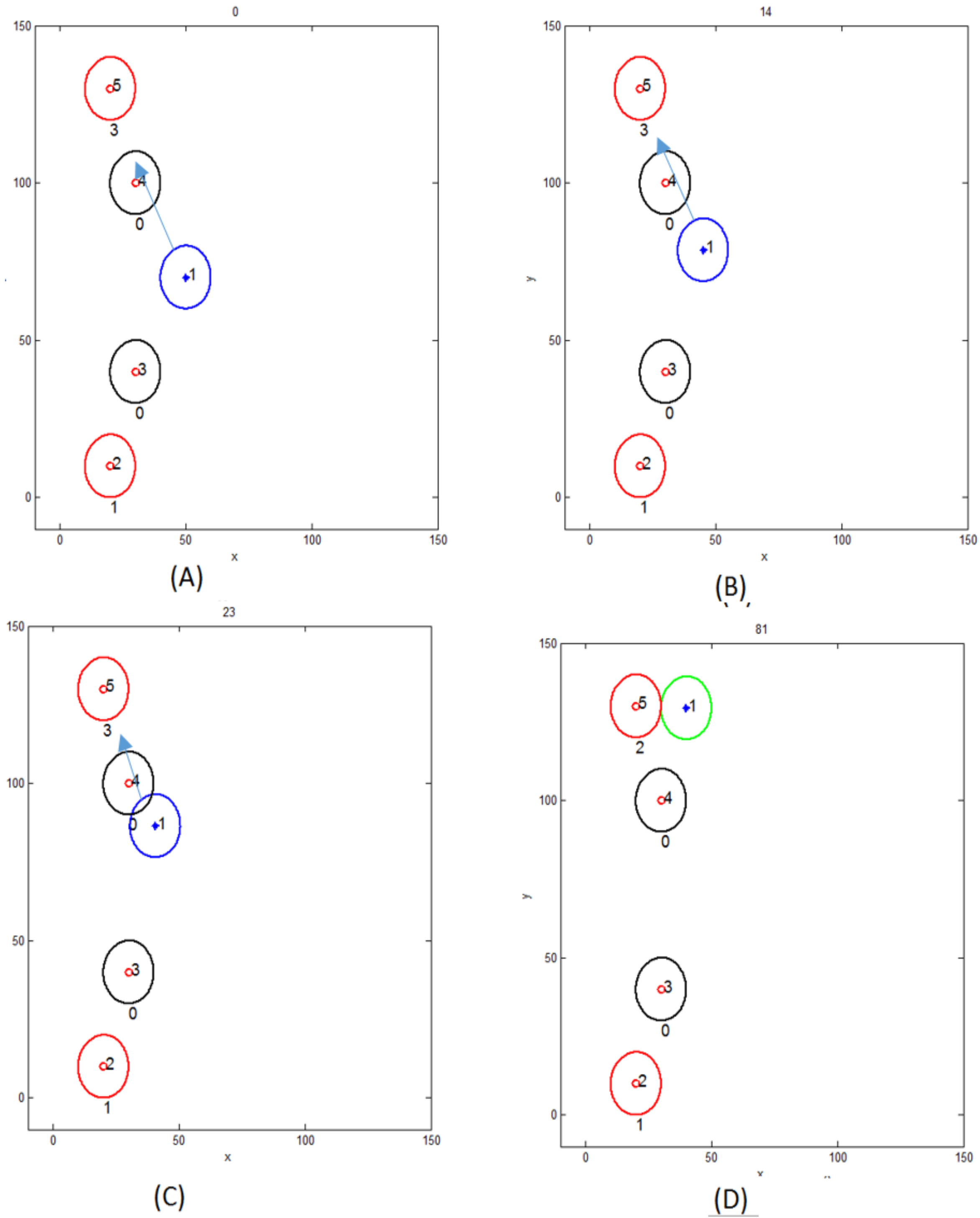


Figure 4.16: An example of how COVER takes longer time and more messages to satisfy landmark demand. The number inside the circle is the number of landmark or robot. The number below the circle is the demand of the landmark. A)Initial position of robot R_1 and landmarks (2-5). B) R_1 moves toward L_5. C) R_1 is getting in range of L_5. D) R_1 associates with L_5 and relocate to a position determined by L_5

is to employ a two-hop communication between the robot R_i and landmark L_j. In COVER approach, when a robot gets associated (i.e. $R_i \in R_a$) or when a landmark demand has been satisfied (i.e. $d(L_j) = 0$), it will collaborate with other landmarks that are still having demand and not satisfied yet (i.e. $d(L_j) > 0$) by applying an attractive force on free robots ($\in R_f$) toward the landmark with the highest demand. This way has many problems: one of them is that the free robot may need more than one iteration until it gets associated with the landmark that it has been attracted toward it. For example, in Figure. 4.16 robot R_1 is getting attracted by landmark L_3 toward L_2 and by L_4 toward L_5. The attraction toward L_5 is higher so it will move toward it. After moving the maximum distance allowed (i.e. 10 meters as in Figure. 4.16-B), R_1 will broadcast a message again to see if there is any change in the environment and to know its neighbors; still the same attractive force will be applied on R_1 toward L_5, so it will continue in the direction of L_5 until it gets in the range of it to associate directly with it as in Figure. 4.16-C. So, in this way we needed many messages at each stop and additional time to collect the messages and process them.

Another problem with COVER is presented in 4.18-A where attractive forces are applied on R_1 from L_3 toward L_2 and from L_4 toward L_5. So, the robot will move one step more as in Figure. 4.18-B, and then the force will be equal from both directions of L_2 and L_5 and the net force will be approximately zero so the robot will not move. In this case, there is no way that COVER can help such robot to associate and at the same time, the robot has already moved some

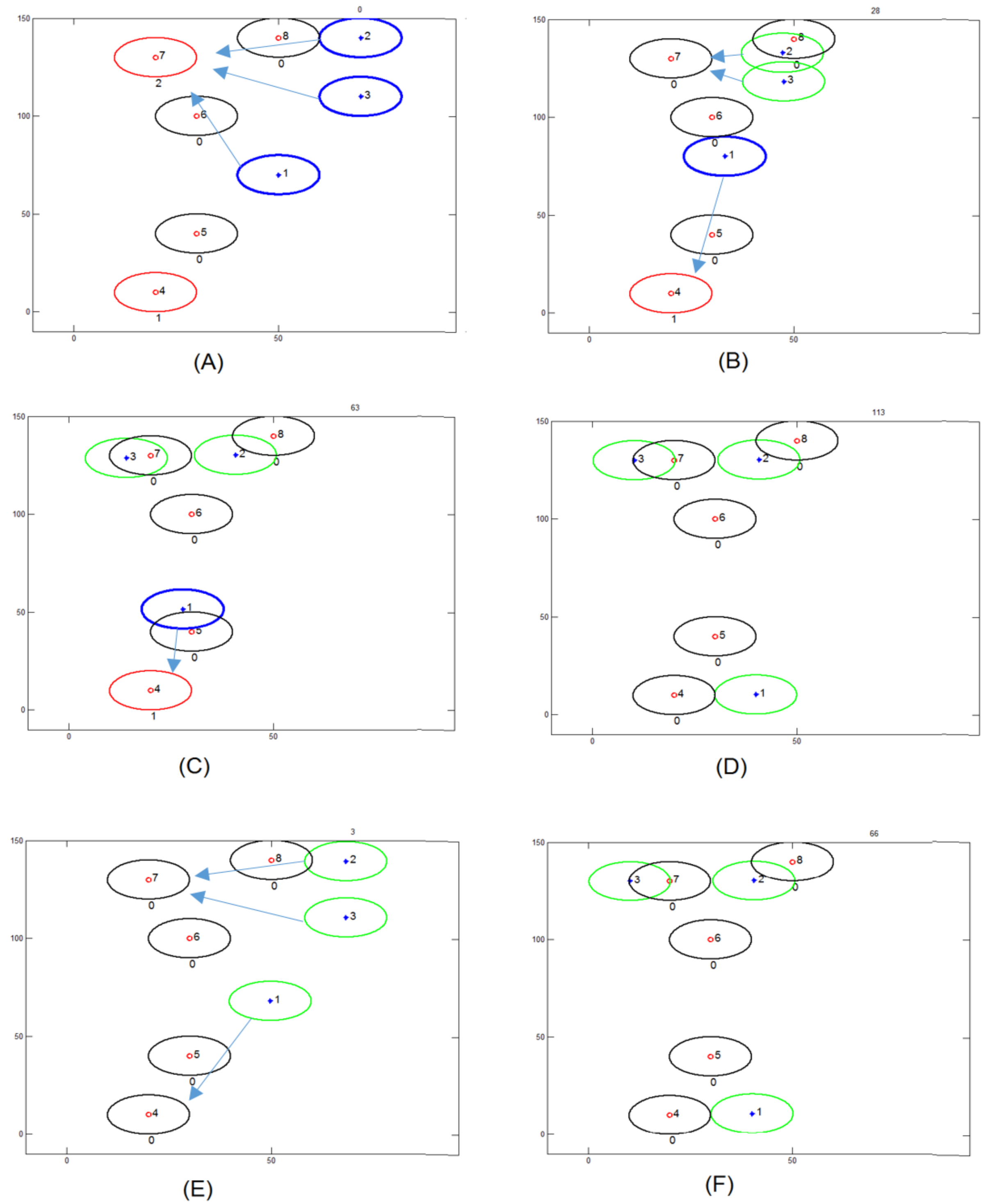


Figure 4.17: A second example of how COVER takes longer time and more messages to satisfy landmark's demand. A) The initial positions of three robots (1, 2, 3) and landmarks (4-8). B) The three robots will move toward L_7. C) R_1 will return toward L_4 D) Each robot is associated with a landmark E) Initial positions of the robots and according to Two-hop approach, all the robots will decide how to move in this step. F) The final deployment according to the Two-hop approach

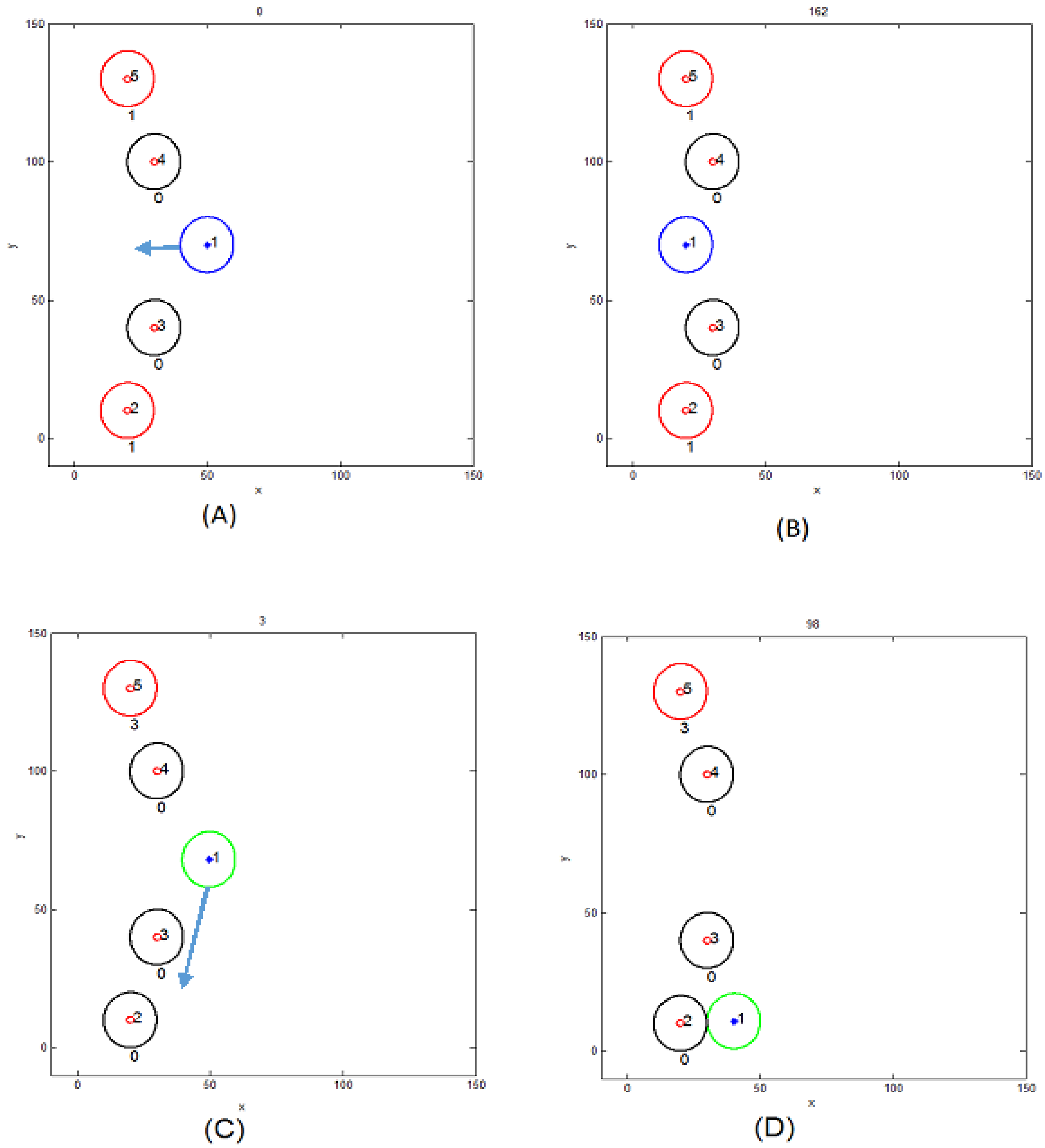


Figure 4.18: An example of how COVER approach failed to make R_1 associates to a landmark and how the two-hop COVER solved the problem. A) The initial position of R_1. B) R_1 can not decide where to move because the force on it is equal from the direction of L_2 and L_5. C) The two-hop COVER helped R_1 to associate with L_2 through L_3 D) R_1 is associated with L_2 and relocated to a position determined by L_2.

distance and sent many messages. So these problems are solved in the Two-hop COVER by changing the role of the satisfied landmarks and associated robots by employing the two-hop communications. They will make a list of the landmarks with demand not satisfied and send it to the free robots near to them on behalf of the landmarks. If the free robot is in the range of one of the landmarks, it will associate with it directly, otherwise, it will send an association request through the one which sent the list (i.e. either a satisfied landmark or an associated robot) as in 4.18-C where R_1 will associate with L_2 though L_3 and so R_1 will not need to send any broadcast messages after it gets associated. That way, we expect that the distance traveled by a robot will be reduced because the robot can decide if it will associate with any landmark without a need to move until it gets in range with that landmark due to the utilization of the two-hop communication. Moreover, the time required to achieve the maximum possible demand will be reduced as well. The second amendment to improve COVER is that when a robot gets associated to a landmark, it will not move immediately toward it, rather, it will stay in its current position until it either finds out that none of its neighboring landmarks has a demand or after multiple iterations (i.e. wait for a certain time). The logic behind this is that initially the robots are close from each other and when one of the robots gets associated, the possibility of being in help for its landmark or other neighbor landmarks is high if this robot stays near to other free robots.

Lemma 4.4 *A robot may encounter a deadlock in COVER.*

Proof.

First, assume that a robot is in a situation where there are two attractive forces from two landmarks. The forces are with the same magnitude but the directions are different, one with angle 45 and one with angle 225. The two forces will cancel each other and the robot will move in the direction of angle 180. Then the robot reaches a place where there will be two attractive forces from two landmarks in two opposite directions as follows:

(Landmark L_1) $\xleftarrow{\overleftarrow{F_{R_1L_1}}}$ (Robot R_1) $\xrightarrow{\overrightarrow{F_{R_1L_2}}}$ (Landmark L_2).

Since $F_{R_1L_1} = w_{L_1} * d_{R_1L_1}$ and $F_{R_2L_1} = w_{L_2} * d_{R_1L_2}$. and $w_{L_1} = demand(L_1) * (number of robots)^{-a}$ and $w_{L_2} = demand(L_2) * (number of robots)^{-a}$. If $demand(L_1) = demand(L_2)$ then $w_{L_1} = w_{L_2}$. Also, if $d_{R_1L_1} = d_{R_1L_2}$ then $F_{R_1L_1} = F_{R_1L_2}$. So, if the forces from two opposite directions, they will cancel each other $\overleftarrow{F_{R_1L_1}} + \overrightarrow{F_{R_1L_2}} = 0$. That will result in a net force of zero and the robot will not move anywhere.

∎

4.4.2 Two-hop COVER Algorithm

The virtual force calculation will be modified to consider the cooperations introduced by Two-hop COVER. The total force will be as in equation 4.18.

$$F_i = \sum_{j=1, j \neq i}^{K} F_{ij} + \sum_{r=1, r \neq i}^{M} F_{ir} \tag{4.18}$$

$j \in N_{R_f}(R_i))$, r is a robot $\in N_{R_a}(R_i)$ and $d(N_l(R_a)) = 0$ or a is a landmark

$\in N_{L_s}(R_i)$ and $d(N_l(L_s)) = 0$.

In computing the composite virtual force, we differentiate between three cases. The first case when the neighboring robots are not associated with any landmark, then, the calculation goes as the basic virtual force [1] based on 4.19.

$$F_{ij} = \begin{cases} w_a(d_{ij} - d_{th}), \theta_{ij} & \text{if } x > d_{th} \\ 0 & \text{if } d_{ij} = d_{th} \\ \frac{w_r}{d_{ij}}, \pi + \theta_{ij} & \text{if } x < d_{th} \end{cases} \tag{4.19}$$

where

$$w_a = \frac{d_{th}}{C_{th}} * (numberofrobot)^{-\alpha} \tag{4.20}$$

and

$$w_r = (numberofrobot)^{\alpha} \tag{4.21}$$

The second case is when the received messages come from associated robots or from landmarks where the demand in their proximity is greater than zero, then the force will be zero and the robot will utitluze two-hop communication to reach out the landmark with demand through either an associated robot as a satisfied landmark.

$$F_{ia} = 0 \tag{4.22}$$

The third case is when the received messages come from associated robots or from landmarks where the demand in their proximity is zero, the calculation goes as follows.

$$F_{ir} = \frac{w_r}{d_{iR}} * \beta \tag{4.23}$$

Two algorithms are presented, one for the operation of the robots (Algorithm 3) and one for the operation of the landmarks (algorithm 4).

4.4.3 Simulation Setup

In order to evaluate the performance of Two-hop COVER, we have conducted extensive simulations for different scenarios. Two main scenarios are presented: one with area size 150m x 150m and the second with 200m x 200m. We have increased the area size to allow for more cooperation to happen and be utilized to see the performance of the two-hop COVER in term of the level of demand satisfaction, time, and distance. In both scenarios, the number of landmarks is 10 and randomly distributed over the area. The number of robots ranges from 15 to 35 randomly distributed over the area. The demand is set to equal the number

Algorithm 3 The algorithm for the operation of the robot in the Two-hop COVER

switch Robot_status **do**
  case R_i unassociated
    R_i sends neighbors position inquiry message
    R_i receives position reply messages from robots ($R_k \in N_r(R_i)$) and landmarks $L_k \in N_l(R_i)$:
    for all replies of landmarks $L_k \in N_l(R_i)$ **do**
      if the demand $d(L_k) > 0$ **then**
        Add the demanding landmark L_k to the potential demanding landmarks list dl.
      else if $d(L_k) == 0$ & the demand of its neighbors $d(L_a) \in N_l(L_k) > 0$ **then**
        add L_a to dl
      else
        add the landmark L_k to the repulsive force list F_r
      end if
    end for
    for all replies of associated robots $R_k \in N_r(R_i)$ **do**
      if its landmark's demand $d(L(R_i)) > 0 \; || \; (d(L(R_i)) == 0$ & the demand of its neighbors $d(L_a) \in N_l(R_k) > 0$ **then**
        add the demanding landmark/s to the potential landmarks list dl.
      else
        add R_k to the repulsive force list F_r
      end if
    end for

Call ATTEMPTASSOCIATION(dl)
if Failed to associate **then**
Compute the composite virtual force (VF) according to equation 4.1
Compute the new position and relocate to it
end if
case associated
if $d(L(R_i)) == 0$ & the demand of its neighbors $d(L_a) \in N_l(R_i) == 0$ $||$ $current\_time - association\_time > threshold$ **then**
relocate to a position determined by its landmark
end if
if receive position-request message **then**
if its landmarks demand $d(L(R_i)) > 0$ **then**
reply with a demand request
else if my neighbors landmarks demand $d(L_a) \in N_l(R_i) > 0$ **then**
Reply with a list of the landmarks that have a demand $d(L_a) > 0$
end if
else if receive an association request **then**
Forward it to the required landmark
else if receive an association accept or reject from a landmark **then**
Forward it to the required robot
else
reply with a repulsive force
end if

Algorithm 4 The algorithm for the operation of the landmark in the Two-hop COVER

for all landmarks $(L_j \in L)$ **do**
    if L_j receives a position request message from a robot **then**
        if $d(L_j) > 0$ **then**
            Reply with its position and demand.
        else if $d(L_j) == 0$ & $d(N_l(L_j)) > 0$ **then**
            Reply with a list of the landmarks that have a demand > 0
        else
            work as a repulsive force
        end if
    else if receive associate message **then**
        if $d(L_j) > 0$ **then**
            Reply with an accept message, and a position to come to it.
        else if the request to a neighbor landmark **then**
            Forward it to the required landmark
        end if
    else if receive accept or reject message from a landmark to a robot **then**
        Forward it to the required robot
    end if
end for

of robots. The simulation parameters are presented in 4.4. Each experiment is repeated until we reach 90% confidence level with all the parameters are randomly generated.

Parameters	**Value**
Simulation tool	Matlab
Number of robots (randomly distributed)	15, 20, 25, 30, 35
Number of landmarks (randomly distributed)	10
Total landmarks demand (randomly distributed)	15, 20, 25, 30, 35
Waiting time	3 sec
Robots transmission range	50 m
Landmarks transmission range	50 m
Area size	150m x 150m, 200m x 200m

Table 4.4: Simulation Parameters of Two-hop COVER

4.4.4 Results and Analysis

The evaluation of the proposed approach is based on four criteria as follows:

1. the percentage of demand satisfaction which measures how many robots succeeded to associate with the landmarks compared to their total demands. The main purpose of this approach is to satisfy as maximum demand as possible with less time, distance, and messages.

2. The total distance traveled by all robots in order to satisfy landmarks' demand: this include the distance moved due to virtual force calculation plus

the distance moved in order to relocate to a specific position determined by a landmark.

3. The total time used to reach the achieved demand satisfaction. This is the time from the start of Two-hop COVER implementation until the robot reaches the position determined by its landmark

4. The total messages exchanged between robots-robots, landmarks - landmarks and robots - landmarks.

Then we compare Two-hop COVER with the original version (COVER) and with a centralized approach. The centralized approach assumes that all the details are known beforehand such as the positions of the landmarks and their demands as well as the position of all robots. These details are not available in our system and that is the motivation behind developing these distributed algorithms.

The results of the first scenario (150m x 150m) are presented in Figure. 4.19, 4.20, 4.21, 4.22.

The percentage of demand satisfaction is presented in Figure. 4.19. We see that the two-hop COVER is able to reach a around 97% level of demand satisfaction compared to the level achieved by the centralized approach. This is due to the utilization of the two-hop communication which allowed the robots to reach to landmarks that are out of their ranges. In addition, since the associated robot stays for a while in its position immediately after getting associated, this helped in utilizing the two-hop communication to satisfy the demand of its landmark or other neighboring landmarks. We see that the two-hop COVER approximately

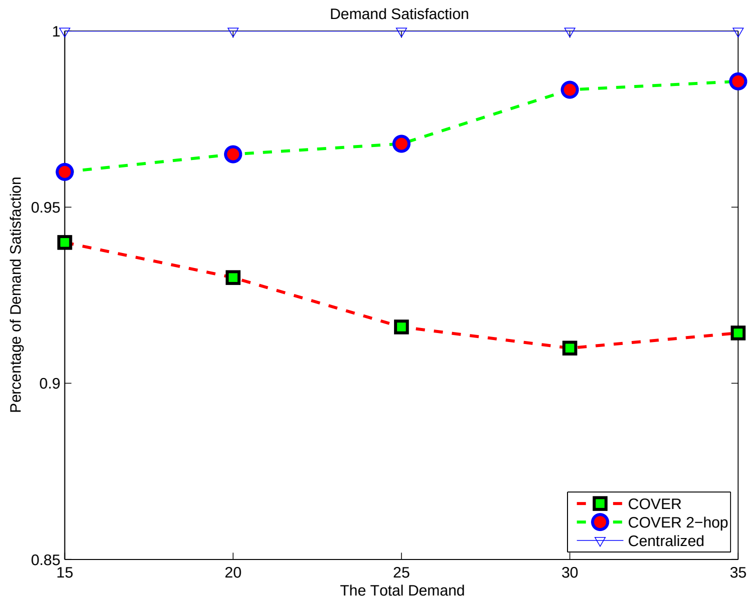


Figure 4.19: The percentage of demand satisfaction, area size = 150m x 150m, the number of robots = the total demand, the communication range= 50 meters

achieves 100% demand satisfaction especially when the number of robots is high compared to the area size and better than the original COVER.

For the total traveled distance, Two-hop COVER succeeded to maintain the total distance traveled by all robots to be the same as COVER, although Two-hop COVER achieved a higher level of demand satisfaction which causes the robot to travel additional distances to relocate to its associated landmark. Instead of making a robot moves some distances in order to discover a landmark and be able to communicate with it, the robot utilized the two-hop communication to reach out the out of range landmarks, but if it succeeds to associate with that landmark, it will eventually travel some distance to get to a specific position determined by its landmark. The total traveled distance is shown in Figure. 4.20.

Another factor is presented in Figure. 4.21 which is the total time needed to reach the achieve level of demand satisfaction in Figure. 4.21 and relocate to the

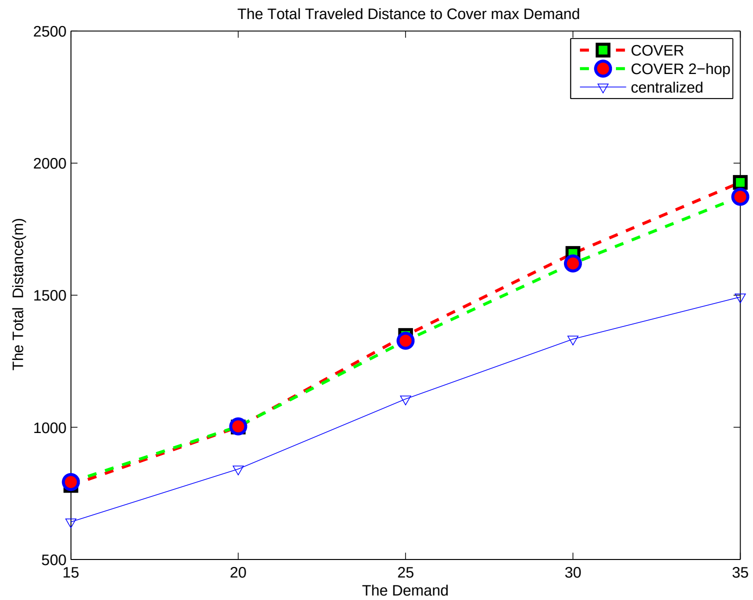


Figure 4.20: The total distance traveled by all robots to reach the achieved level of demand satisfaction, area size = 150m x 150m, the number of robots = the total demand, the communication range= 50 meters

position determined by the landmarks. Although we introduce a waiting time by the associated robots if their landmark's demand is still not satisfied, the two-hop COVER is able to reduce the total time compared to the original COVER by around 20-30% because Two-hop COVER uses two-hop communication which reduces the main time used to search for a landmark. Also, the two-hop COVER needed around 40% increase in the total time compared to the centralized approach. The percentage of improvements of Two-hop COVER over COVER is shown in Table 4.5. We can see that Two-hop COVER reduces the total time by around 16% when the number of robots is 15 and by 27% when the number of robots is 35. This happens because increasing the number of robots will increase the chances of using two-hop communication between robots and unsatisfied landmarks. Also, we can see that Two-hop COVER takes around 50% increase in the

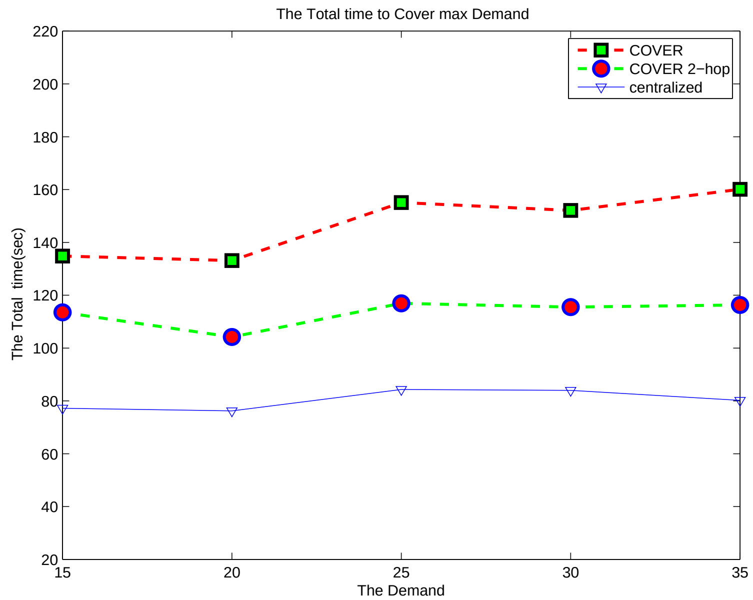


Figure 4.21: The total time needed to reach the achieved level of demand satisfaction, area size = 150m x 150m, the number of robots = the total demand, the communication range= 50 meters

total time compared to the centralized approach.

Robots	**Centralized**	**COVER**	**Two-hop**	**% of Two-hop VS. Centralized**	**% of Two-hop VS. COVER**
15	77.20	134.80	113.50	+47%	-16%
20	76.20	133.10	104.20	+37%	-22%
25	84.30	155.10	116.90	+39%	-25%
30	84.00	152.10	115.50	+38%	-24%
35	80.20	160.10	116.30	+45%	-27%

Table 4.5: The total time in seconds for each approach for each number of robots. Then we show the percentage of improvements of the Two-hop COVER compared to the centralized and to COVER

The total number of messages is the last factor to show in this study. Although the two-hop COVER uses a two-hop association, it succeeded to reduce the total messages exchanged by around 40-50% because the faster the robot gets associated, the less number of messages will be used especially the virtual force

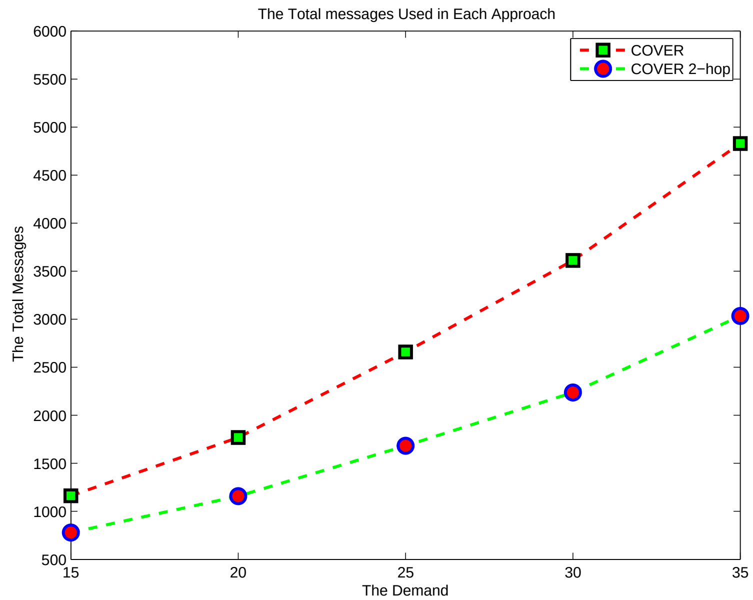


Figure 4.22: The total messages exchanged till the end of implementing Two-hop COVER, area size = 150m x 150m, the number of robots = the total demand, the communication range= 50 meters

messages and the cooperation messages. The associated robots will not broadcast any virtual force messages and consequently, no replies will be needed. This way the total messages are reduced as in Figure. 4.22.

Increasing the area size will affect the performance of the original COVER as well as the Two-hop COVER. However, the Two-hop COVER maintained a high level of performance compare to the original COVER and close to the centralized approach. When the area of interest is wide, this means that the mission of locating unsatisfied landmark is more difficult. The two-hop communication will be in help in such a scenario and will reduce the total time and distance to reach the maximum possible demand.

We see in Figure. 4.23 that the level of demand satisfaction achieved by the two-hop COVER is around 90% of the centralized approach. Since the area size is

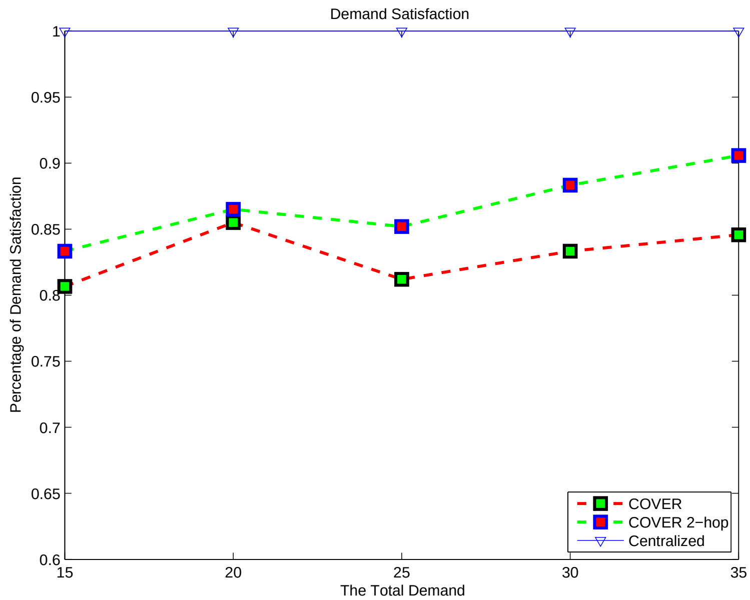


Figure 4.23: The percentage of demand satisfaction, area size = 200m x 200m, the number of robots = the total demand, the communication range= 50 meters

high, some landmarks are far from the robots and even the two-hop communication could not reach them, so that why it could not reach 100% demand satisfaction.

The total traveled distance increased by around 30% compared to the previous scenario. The reason behind this is that the robot will not easily locate a landmark with a demand. Still the total distance traveled by all robots is almost the same to that of the original COVER although the demand satisfaction of the two-hop COVER is higher which implies that some more robots will travel additional distance to relocate to the positions determined by their landmarks. Moreover, in Figure. 4.24 we can see that the total distance of the Two-hop COVER increases by around 30% compared to that of the centralized approach.

Additionally, the total time is also increased for the Two-hop COVER by around 15% compared to the previous scenario, but it is still less than the original COVER and it reduced the time of COVER by around 30% as in Figure. 4.25.

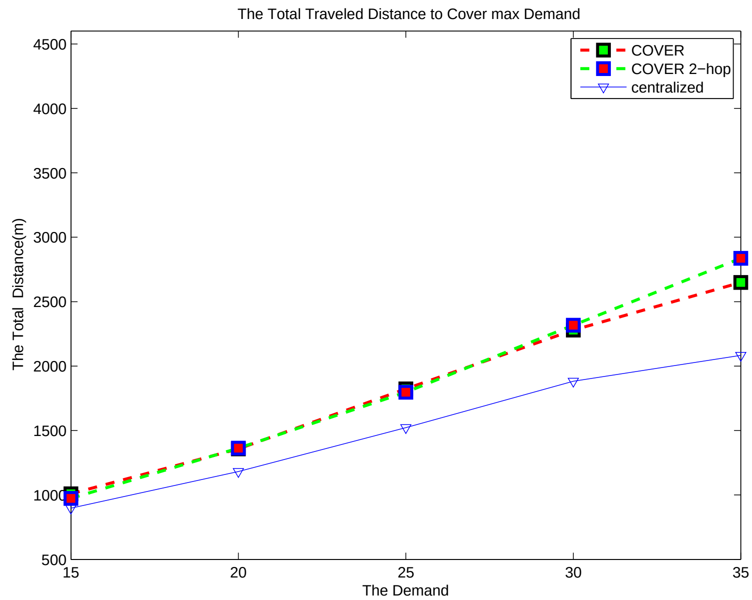


Figure 4.24: The total distance traveled by all robots to reach the achieved level of demand satisfaction, area size = 200m x 200m, the number of robots = the total demand, the communication range= 50 meters

The cause of the increase is the introduced waiting time that some robots needed to utilize to help their landmarks get their demands. This waiting time was used in this scenario because some landmarks were not able to get their demand early at the deployment because of the increased size of the area which caused that delay.

And finally the same still holds for the total number of messages exchanged as in Figure. 4.26. The Two-hop COVER was able to reduce the total number of messages by around 30% compared to the original COVER.

Finally, We have also studied the effect of using different transmission ranges. As we have seen, all the above scenarios were with a transmission range of 50 meters. In this section, we will show the effect of increasing the transmission ranges from 50 meters to 80 meters. First, in Figure. 4.27 we see that increasing

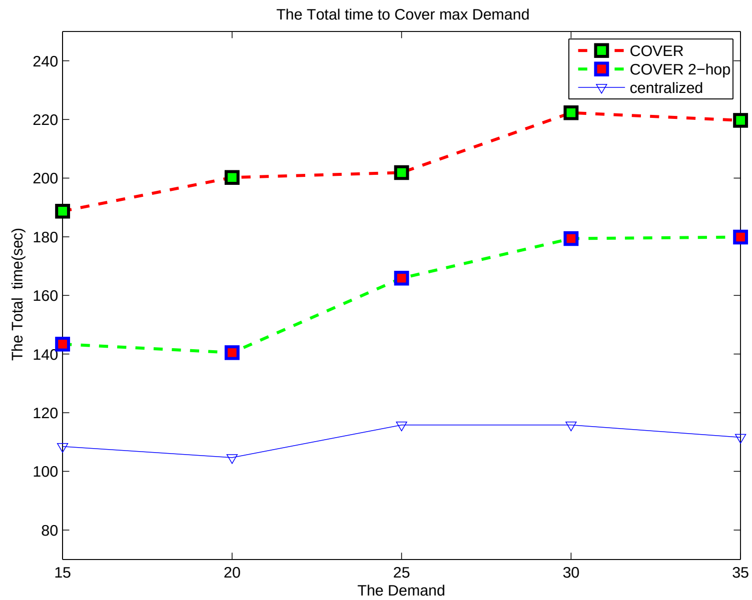


Figure 4.25: The total time needed to reach the achieved level of demand satisfaction, area size = 200m x 200m, the number of robots = the total demand, the communication range= 50 meters

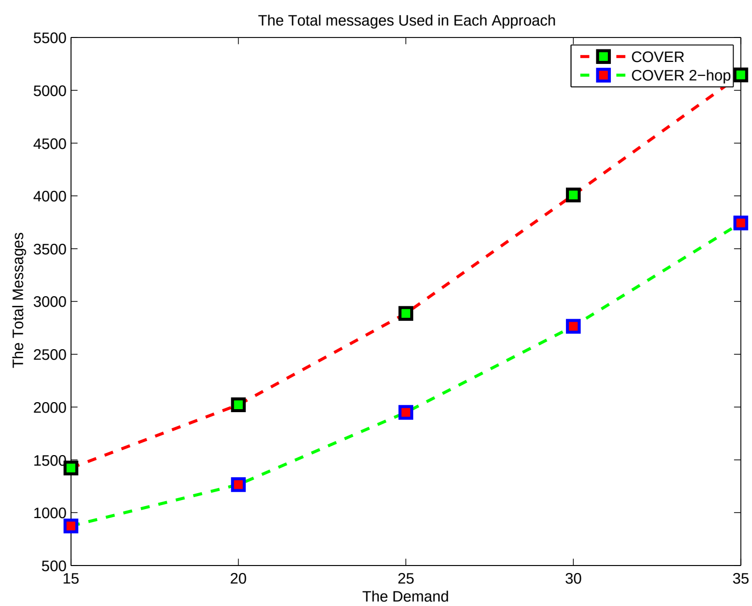


Figure 4.26: The total messages exchanged till the end of implementing the two-hop COVER, area size = 200m x 200m, the number of robots = the total demand, the communication range= 50 meters

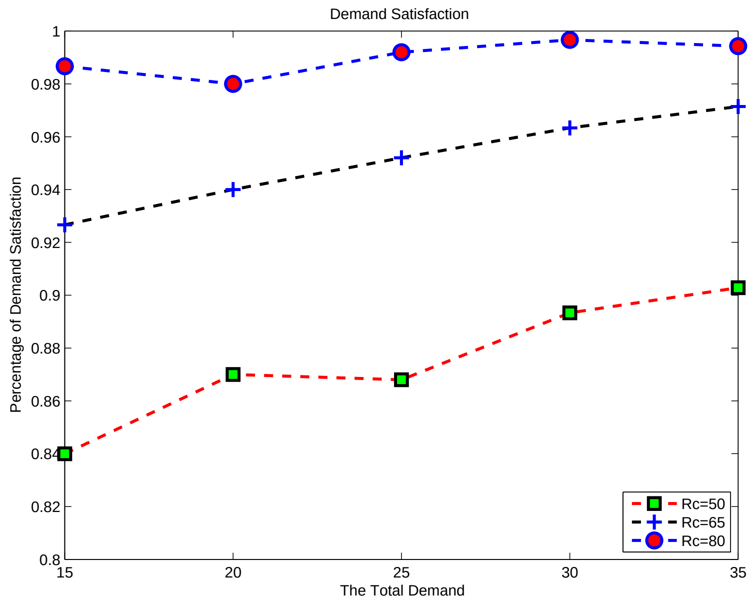


Figure 4.27: The percentage of demand satisfaction, area size = 200m x 200m with different transmission range

the transmission range helps in improving the demand satisfaction until it almost reaches 100% demand satisfaction when the transmission range is 80. The higher the transmission range, the easier it becomes for a robot to locate a landmark and associate to it. Also, the cooperation scale will increase and each associated robot or satisfied landmark will be able to help other unsatisfied landmarks in their new transmission range which becomes wider as we increase the transmission range. Table 4.6 show the percentage of improvements due to increasing the communication range. We see that increasing the communication range from 50m to 65m improves the demand satisfaction by around 10% and increases the communication range to 80m improves the demand satisfaction by around 17% when the number of robots is 15 and by 10% when the number of robots is 35.

Moreover, the total distance remains the same when the demand is less than 25, but when the demand increases the total distance decreases by around 10%

Robots	Rc=50	Rc=65	%improvements	Rc=80	%improvement
15	0.84	0.93	10%	0.99	17%
20	0.87	0.94	8%	0.98	13%
25	0.87	0.95	10%	0.99	14%
30	0.89	0.96	8%	1.00	12%
35	0.90	0.97	8%	0.99	10%

Table 4.6: The percentage of demand satisfaction for different communicatin ranges in Two-hop COVER. We show the improvement from Rc=50 to Rc=65 and 80.

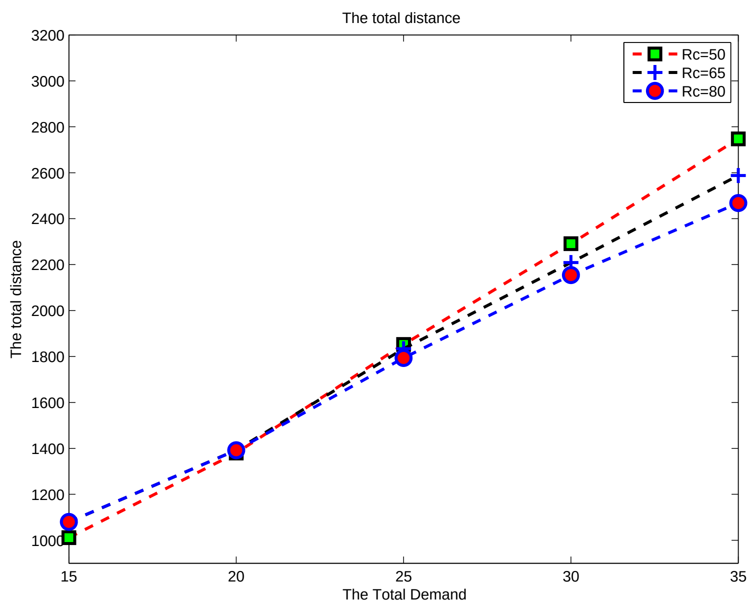


Figure 4.28: The total distance traveled by all robots to reach the achieved level of demand satisfaction, area size = 200m x 200m with different transmission range

with the increase in the transmission range. Although the increase in the demand satisfaction achieved means that some robots will need to travel more distance to relocate to the positions determined by their landmarks, the overall traveled distance of all robots is reduced. Each robot will not need to move more distance to locate a landmark. The higher transmission range will help in finding landmarks and so minimizing the total distance needed to search for a landmark as we see in Figure. 4.28.

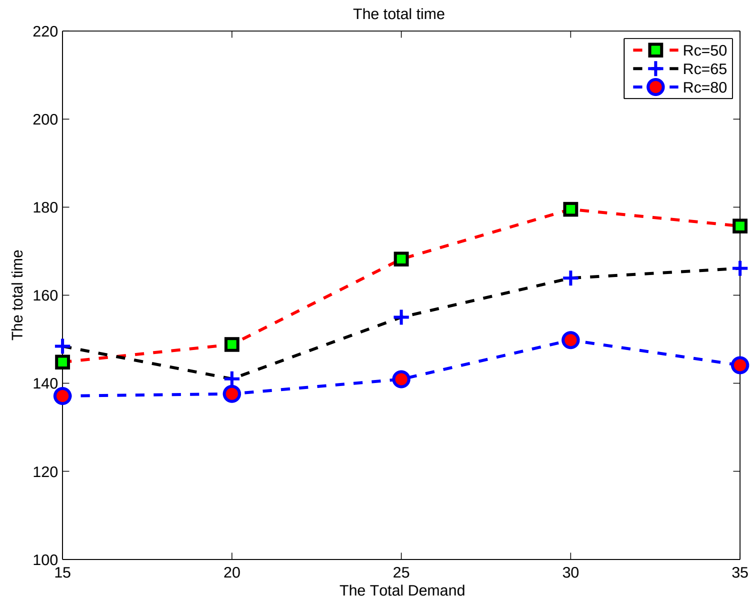


Figure 4.29: The total time needed to reach the achieved level of demand satisfaction, area size = 200m x 200m with different transmission range

For the total time needed to achieve the required deployment, the increase in the transmission range will reduce the total time needed as in Figure. 4.29. For example, increasing the transmission range to 80m reduces the total time by around 15-30% compared to 50m transmission range. Although we compute the time till the robots reach the position determined by their landmarks which means more time when the demand satisfaction is high, the overall deployment time is shortened. The same holds for the total messages where the increase in the transmission range will let the robots get associated so quickly to landmarks and consequently reduces the total messages as well as in Figure. 4.30.

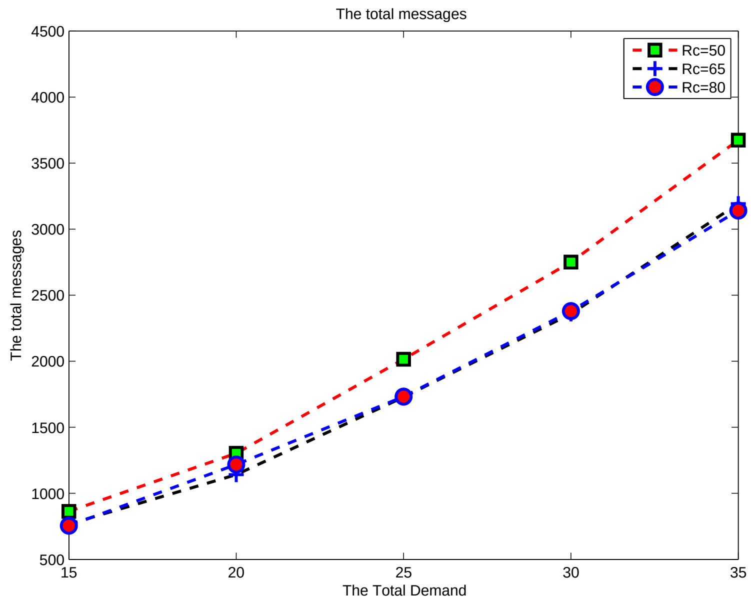


Figure 4.30: The total messages exchanged till the end of implementing the two-hop COVER, area size = 200m x 200m with different transmission range

4.5 Trace Fingerprint

4.5.1 Introduction

We have presented two versions of COVER, the original COVER, and the Two-hop COVER. However, we still see that even though the level of demand satisfaction sometimes reaches around 95% and more in some scenarios where area size is not large, it fails to reach more than 90% demand satisfaction when the area size gets large such as in the previous section when the area size is 200m x 200m. In this section, we present a modification to the Two-hop COVER that can guarantee 100% demand satisfaction given that the number of robots is enough to meet all landmarks demand. One possible solution is to let the remaining robots wander the whole area randomly until they locate a landmark with demand. This approach could take long time to satisfy the demand and may not always reach

100% demand satisfaction. If the area size is large, the robot may not go to the region of demand. Additionally, if the total demand is less than the available robot, then there will be no demand for some of the robots. In this case, they will not find out that there is no need from them and they will keep moving randomly until they deplete their energy. Moreover, the random movements could result in a very high cost in term of the traveled distance and the time needed to locate unsatisfied landmark. So, in order to reduce such randomness and guide the remaining robots throughout the area till they find an unsatisfied landmark, we propose that each robot will utilize its history to visit only the locations that it has not visited before. Moreover, each robot will communicate with its neighboring robots and landmarks to use their current positions and history, if applicable, to guide its movements. After applying the Two-hop COVER, if a robot fails to associate and the following conditions are satisfied, the robot moves to implement Trace Fingerprint introduced here. The conditions are : 1) There is no force applied on a robot for many iterations (i.e. 10 iterations depends on system parameters like the area size and the speed of the robots). 2) The only force exerted toward the robot is a repulsive force for many iterations (i.e. 15 iterations), then the robot moves to implement Trace Fingerprint.

Figure 4.31 shows an example of how to use Trace Fingerprint. In Figure. 4.31-B, according to Two-hop COVER, robot R_{14} is unable to locate an unsatisfied landmark, so it gets stuck at its position with no force applied on him to be directed to move so it will start the implementation of Trace Fingerprint. In

Figure. 4.31-C, R_{14} will divide the area into squares and compute the coverage of each square based on the history it has built, the covered area is shown with canyon color. Then the robot will move toward the nearest partially covered square which is to the right as in Figure. 4.31-D. Once the robot reaches, it will search for a landmark but it did not find any so it will collect the traces of the new neighbors and repeat the same process. The coverage level computation is repeated again and the robot finds the nearest uncovered area and moves toward it. The process is repeated until the robot gets close the demanding landmark (L_{16}) as in Figure. 4.31-E and will associate to it. Then finally the robot will relocate to a position determined by its landmark as in Figure. 4.31-F.

4.5.2 Algorithm

Each robot will maintain a trace set $t_i = \{(x_n, y_n), n = 1...N\}$ where N is the number of points that the robot stopped at and checked for the presence of landmarks and did not find any. Once a robot gets stuck, it will collect the traces of the other robots and build a new one $T_i = \bigcup_{\forall i}\{t_i\}$ where $i \in \{N_R(R_i)$ and $N_L(R_i)\}$. Each point in the trace will be used to build a virtual map of the places that have been already visited so $\forall j \in T$, it will be represented in the virtual map as $VM_j =$ a circle centered at (x_j, y_j) with radius R_c and each point within the circle means that there is no landmark on it. The virtual map $VM = \bigcup_{\forall j}\{VM_j\}$ where $j \in T_i$. Then, the robot will choose the nearest point not covered in VM and move toward it. Once it reaches its target it will check for the presence of a

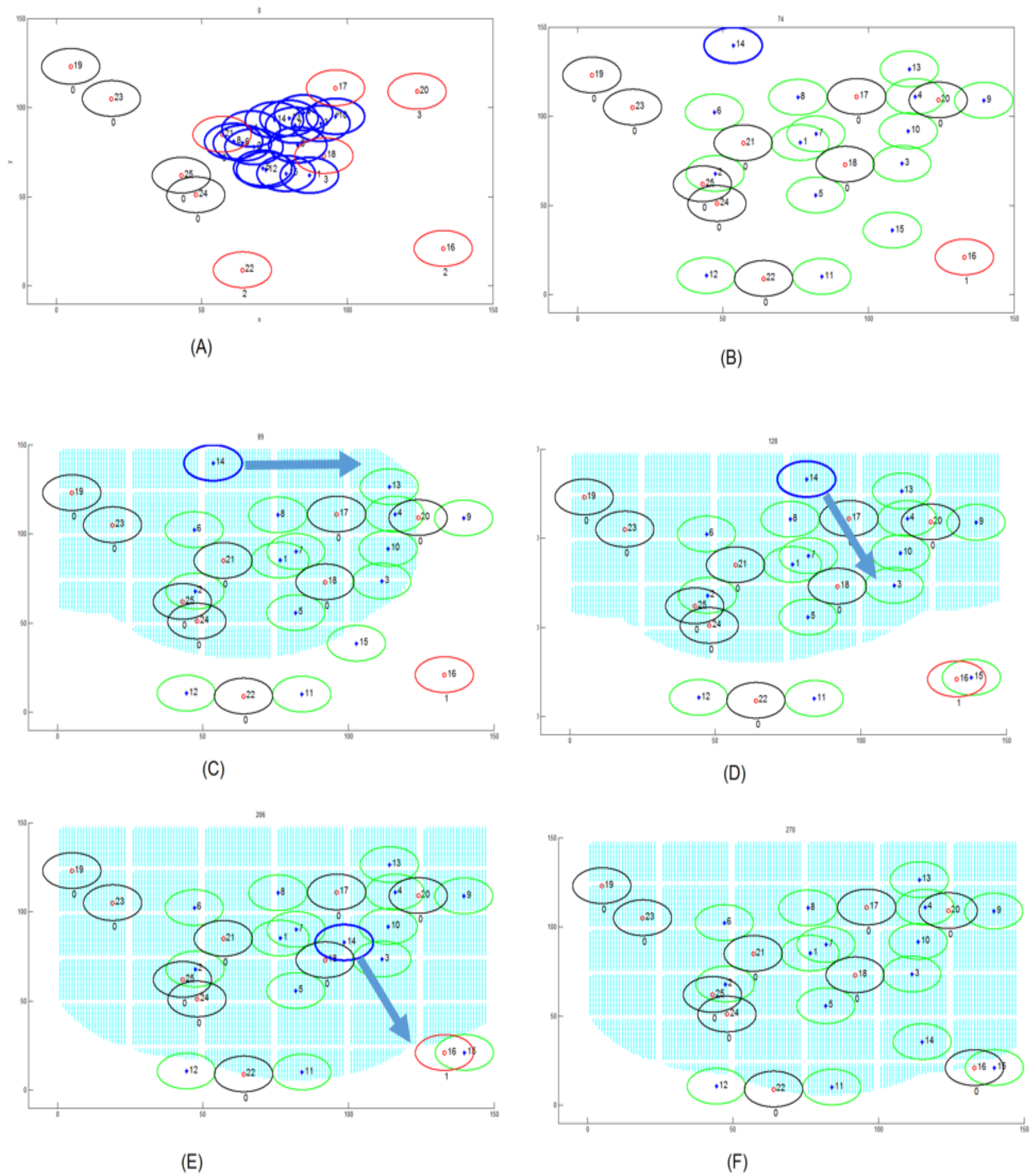


Figure 4.31: An example of how Trace Fingerprint guarantees 100% demand satisfaction. (A)The initial deployment of the robots. (B) the final deployment according to two-hop COVER. (C) The start of Trace Fingerprint implementation by robot R_{14}. (D) Robot R_{14} moves toward the nearest free area to the right. (E) Robot R_{14} moves downward toward the nearest free area and gets in range of the landmark robot L_{16}. F) Robot R_{14} gets associated to landmark L_{16}

landmark with a demand, if it fails, it will repeat the same process. Algorithm 5 presents the additional steps that will be implemented by a robot when it reaches the stage that the Two-hop COVER is not helping it to get associated with a landmark.

4.5.3 Results and Analysis

We have tested the proposed Trace Fingerprint using the same parameters in Table 4.4 using the area size of 200m x 200m. We compare Trace Fingerprint with the Random Waypoint (RWP) approach. Since the RWP represents a natural movement for the robot searching for the landmark, we consider RWP as our baseline approach. We compare them in term of the number of robots that are able to satisfy the demand of the landmarks, the total traveled distance of robot, the total time, and the total number of messages. So, the two-hop COVER will be applied first, if the robot fails to find a landmark with demand, it will apply Trace Fingerprint or the RWP until it locates a landmark and associate to it.

We can see in Figure. 4.32 that Trace Fingerprint succeeded to reach 100% demand satisfaction. The same level of demand satisfaction was also achieved by the RWP, however, the difference will be shown in Figure. 4.33, 4.34, 4.35 regarding the total time, distance, and messages.

Trace Fingerprint needs additional movements in order to locate landmarks but they are small compared to that needed for the RWP as shown in Figure. 4.33. While Trace Fingerprint caused around 20-30% increase in the distance to

Algorithm 5 The algorithm for the operation of the free robot in Trace Fingerprint

```
switch Robot_status do
    case unassociated
        if total force ==0 or all force types== repulsive then
            count++
            if count > α then
                change robots_status to unassociated_searching
                get the history of the neighboring robots and the position of the landmarks and add them to the history
            end if
        else
            add my next position to the history
        end if
    case unassociated_searching
        repeat
            divide the area into squares
            Compute the coverage level of each square based on the history
            make a list of squares that are not fully covered
            choose the nearest square and move toward it for a distance d
            if gets associated then
                change robots_status to associated
            else
                gets the positions of the neighbors landmarks and robots and update its history.
            end if
        until gets associated
```

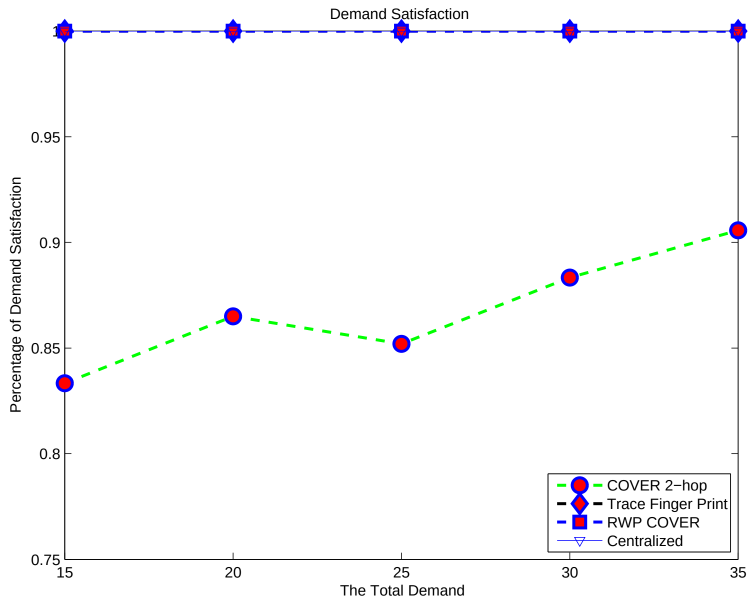


Figure 4.32: The percentage of demand satisfaction of Trace Fingerprint, area size = 200m x 200m, communication range=50m, the number of robots= the demand

reach 100% demand satisfaction compared to the Two-hop COVER, RWP caused more than 100% increase in the distance traveled. This shows the effectiveness of the proposed approach to guide the robots' movements rather than making them move randomly. In the Trace Fingerprint, the robot will not visit the previously visited places by him or by its neighbor robots which increase the chances of locating landmarks quickly and reduces the total traveled distance. Additionally, in Table 4.7 show the percentage of distance needed by the Trace Fingerprint and RWP compared to the centralized approach. We see that Trace Fingerprint needs to travel more distance around 70% compared to the centralized, however, the RWP needs more than 150% when the number of robots is 15 and 100% when the number of robots is 35.

Moreover, due to the guided movements in Trace Fingerprint, the total time needed to reach 100% demand satisfaction is around half the time needed for RWP

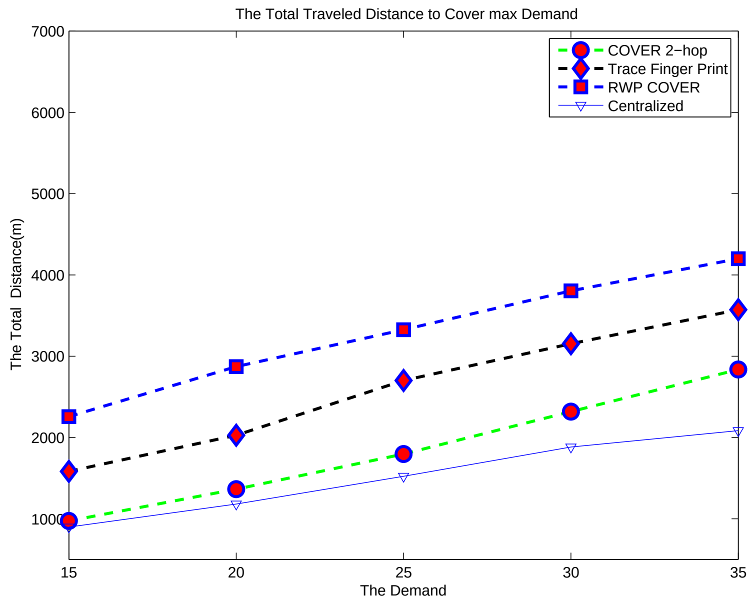


Figure 4.33: The total distance of Trace Fingerprint to achieve the maximum demand satisfaction, area size = 200m x 200m, communication range=50m, the number of robots= the demand

Robots	**Centralized**	**Fingerprint**	**%**	**RWP**	**%**
15	897.60	1582.30	76%	2256.70	151%
20	1181.50	2026.30	72%	2871.70	143%
25	1522.50	2701.50	77%	3325.10	118%
30	1882.50	3153.40	68%	3804.10	102%
35	2083.60	3572.30	71%	4199.80	102%

Table 4.7: The total traveled distance of Trace Fingerprint compared with the centralized and RWP. The percentages are calculated based on the difference from the centralized approach

as shown in Figure. 4.34.

Finally, the longer it takes to reach the maximum demand satisfaction, the higher the number of messages that will be used. So, we see that RWP uses more messages compared to Trace Fingerprint as in Figure. 4.35.

The better performance of Trace Fingerprint compared to RWP comes at a computational cost. Each free robot needs to make calculations that are not needed in RWP. The robot needs to store traces its previously visited locations

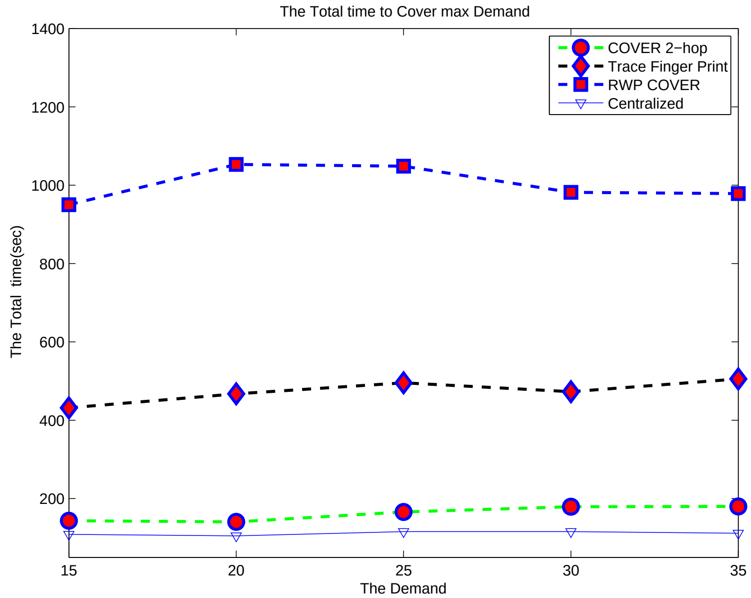


Figure 4.34: The total time of Trace Fingerprint to achieve the maximum demand satisfaction, area size = 200m x 200m, communication range=50m, the number of robots= the demand

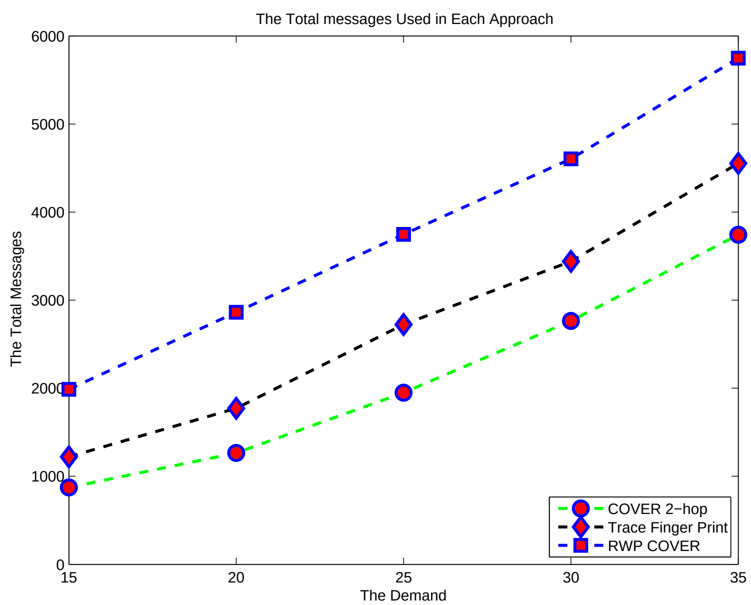


Figure 4.35: The total number of messages of Trace Fingerprint to achieve the maximum demand satisfaction, area size = 200m x 200m, communication range=50m, the number of robots= the demand

and the locations of its neighbors. Moreover, at each iteration, the robot needs to find the coverage level of each square in order to decide its next move. So, for time critical applications and with the current advance in the computational resources, we can ignore such computational overhead and consider only the other factors (time, distance, and messages).

4.6 COVER and Two-hop COVER Approaches with Fairness

4.6.1 Introduction

Other versions of COVER and Two-hop COVER are proposed here. Their main aim is to address the scenarios when the collective demand of the landmarks is greater than the available robots. In this case, we don't want the algorithm to work greedy as in the previous versions. In the previous versions, COVER and Two-hop COVER, the landmarks will only cooperate with other landmarks when their remaining demand is zero. So, we would like to distribute the robots in a way such that no landmark gets all of its demand while the other landmarks around it having their demand not satisfied to a certain level. We achieved this fairness by proposing a minimum demand satisfaction level (min_{DS}). When a robot reaches the $min_D S$ it will start to cooperate with its neighboring landmarks that are having demand greater than the current landmark's demand. In this way, each landmark will secure its minimum level of demand satisfaction and in the same time help other landmarks getting part of their demand. In order to study the level of fairness achieved, we show results of Jain's fairness index for the remaining demand of the landmarks in the proposed approach and compare it with the greedy version of COVER or Two-hop COVER according to equation 4.24. The minimum satisfaction level (min_{DS}) is set to be equal to 50% of the original demand.

$$J = \frac{(\sum_{i=1,m=1}^{N}(x_i - x_m))^2}{N\sum_{i=1,m=1}^{N}(x_i - x_m)^2} \tag{4.24}$$

4.6.2 Fairness-aware COVER

So, we start by studying Fairness-aware COVER. We see that the employed fairness approach succeeds to reduce the difference of the remaining demand of all the landmarks. We measure this factor by calculating Jain's fairness of the remaining demand that is shown in Figure 4.36. We can see that Fairness-aware COVER is able to improve Jain's fairness. Additionally, we measure the standard deviation of the remaining demand which gives almost the same conclusion as shown in Figure 4.42.

For the other factors, we see that the Fairness-aware COVER takes slightly higher distance (less than 10% increase in the total distance), and a substantial increase in the total number of messages (around 10-50% increase) compared to COVER. This is because when a robot has to decide between more than one landmark, it will choose the one with the highest demand, not the closest one

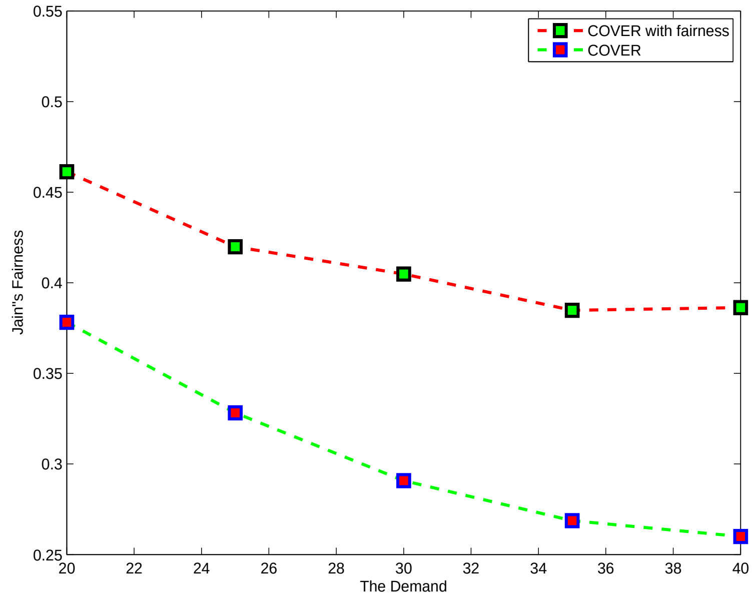


Figure 4.36: Jain's fairness index of the remaining demand of the landmarks for Fairness-aware COVER. The number of robots=demand-10, area size=150m x 150m, communication range=50m

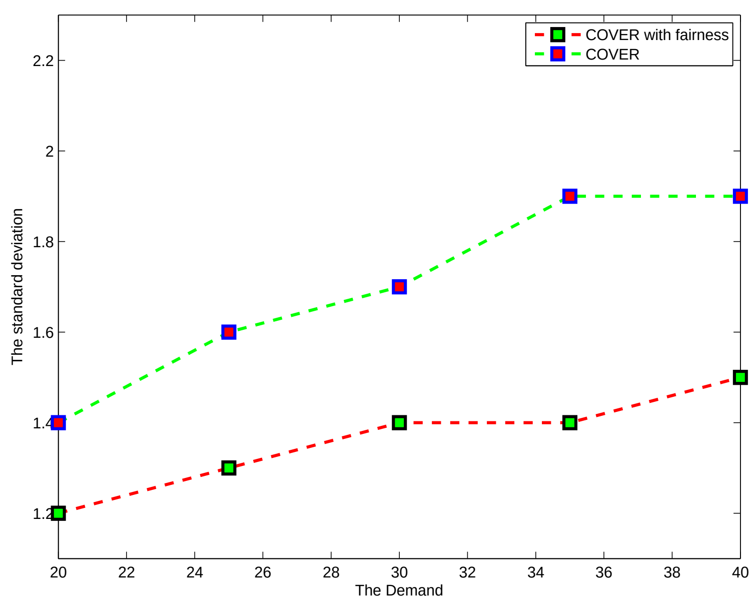


Figure 4.37: The standard deviation of the remaining demand of the landmarks for Fairness-aware COVER.The number of robots=demand-10, area size=150m x 150m, communication range=50m

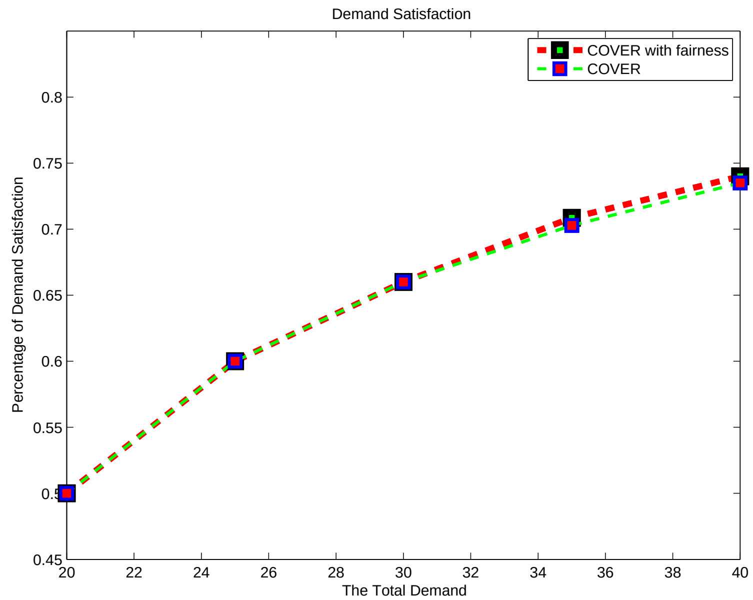


Figure 4.38: The level of demand satisfaction for Fairness-aware COVER. The number of robots=demand-10, area size=150m x 150m, communication range=50m

which will increase the total distance as shown in Figure 4.39. Moreover, if there is a close landmark with a demand but this landmark has other neighbors with a demand greater than its demand, it will attract the robot toward one of them. This will result in a higher distance. Messages as well will increase in this version because the cooperation will start earlier than before. Each landmark will start to cooperate once it secures its min_{DS} which will yield a higher number of messages, sometimes it reachs around 50% when the demand is 40 as shown in Figure 4.41. The higher the level of demand, the high number of messages will be exchanged to help other landmarks that are having demand. The same demand satisfaction has been achieved by the two approaches as shown in Figure 4.38.

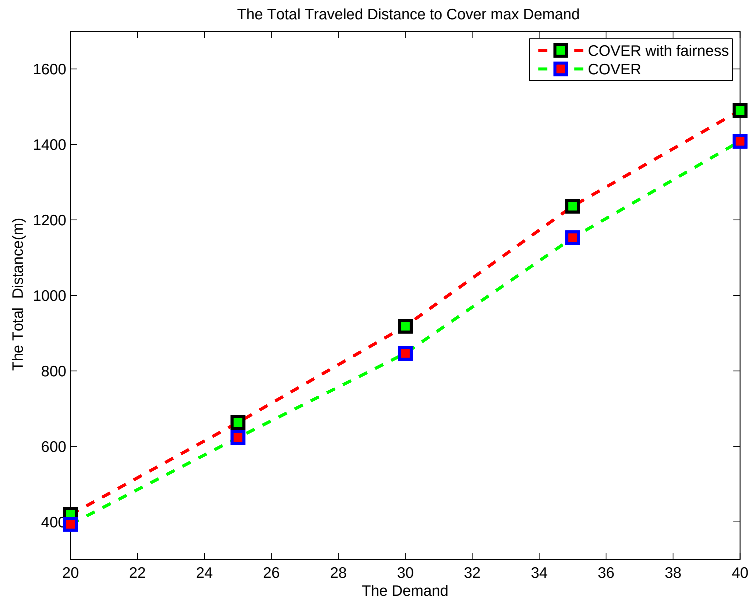


Figure 4.39: The total distance traveled to achieve the maximum possible demand satisfaction in Fairness-aware COVER. The number of robots=demand-10, area size=150m x 150m, communication range=50m

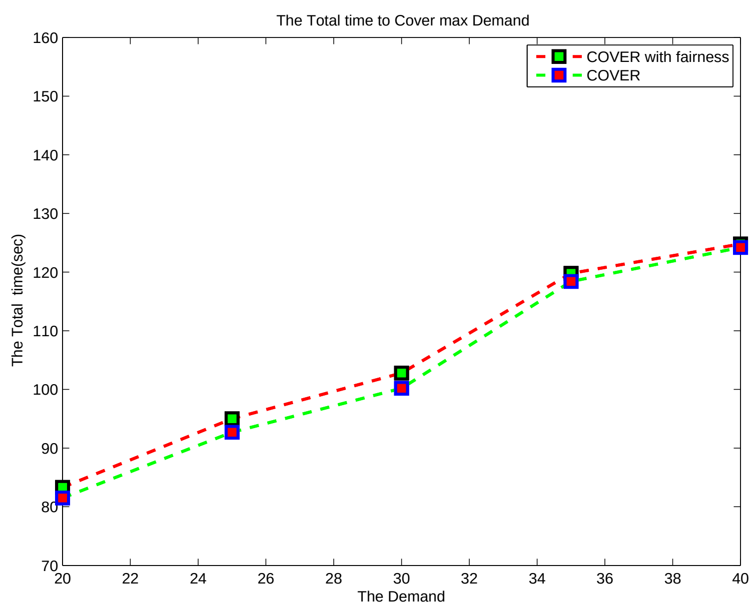


Figure 4.40: The total time traveled to achieve the maximum possible demand satisfaction in Fairness-aware COVER. The number of robots=demand-10, area size=150m x 150m, communication range=50m

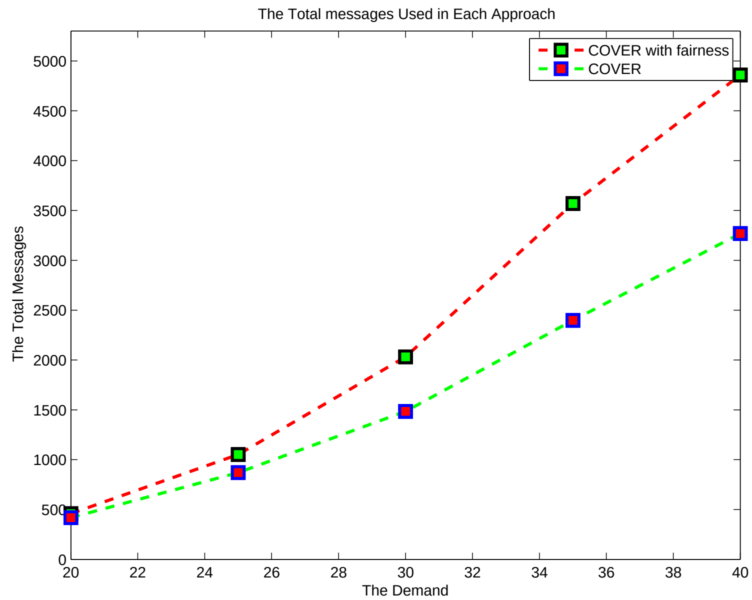


Figure 4.41: The total number of messages for Fairness-aware COVER. The number of robots=demand-10, area size=150m x 150m, communication range=50m

4.6.3 Fairness-aware Two-hop COVER

Then we study Fairness-aware Two-hop COVER to show its effectiveness. We first measure Jain's fairness index as shown in Figure 4.43. We see that Jain's fairness in Fairness-aware Two-hop COVER is close to 1 compared to Two-hop COVER. Figure 4.42 shows the standard deviation of the remaining demand . We see that Two-hop COVER has a larger deviation compared to the Fairness-aware Two-hop COVER. One more thing is that the new approach did not get affected by the increase in the demand.

However, the introduction of fairness affected other metrics such as the total time, distance, and messages. The level of demand satisfaction does not change as in Figure. 4.44. But the total distance increases in the fairness-aware version by around 20% as in Figure. 4.45. The increase in the total distance is due to the

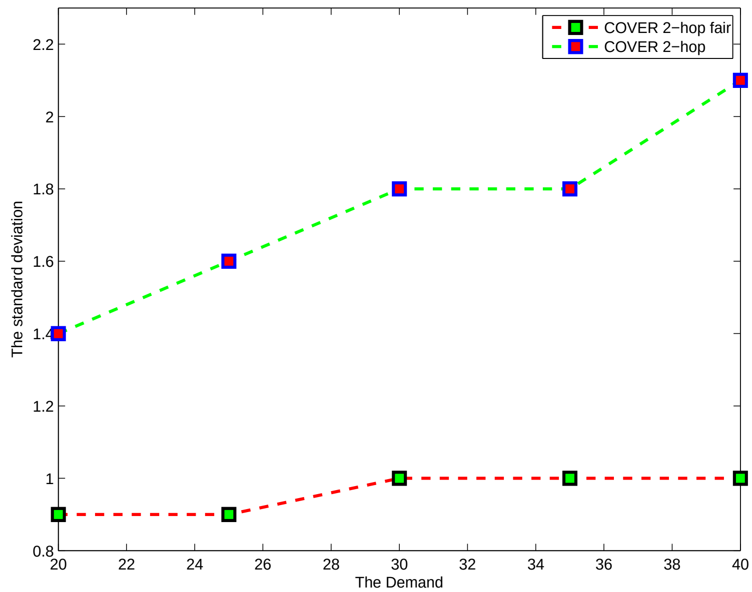


Figure 4.42: The standard deviation of the remaining demand of the landmarks for Fairness-aware Two-hop COVER. The number of robots=demand-10, area size=150m x 150m, communication range=50m

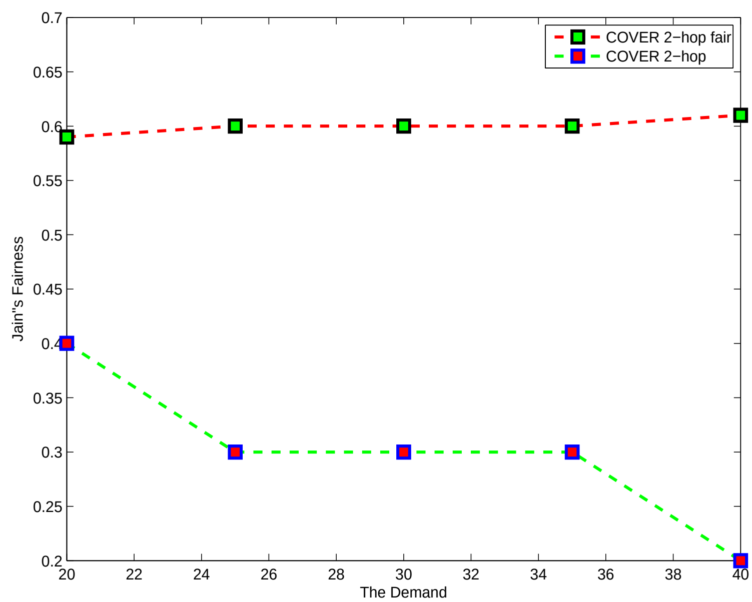


Figure 4.43: Jain's fairness index for the remaining demand of the landmarks for Fairness-aware Two-hop COVER.The number of robots=demand-10, area size=150m x 150m, communication range=50m

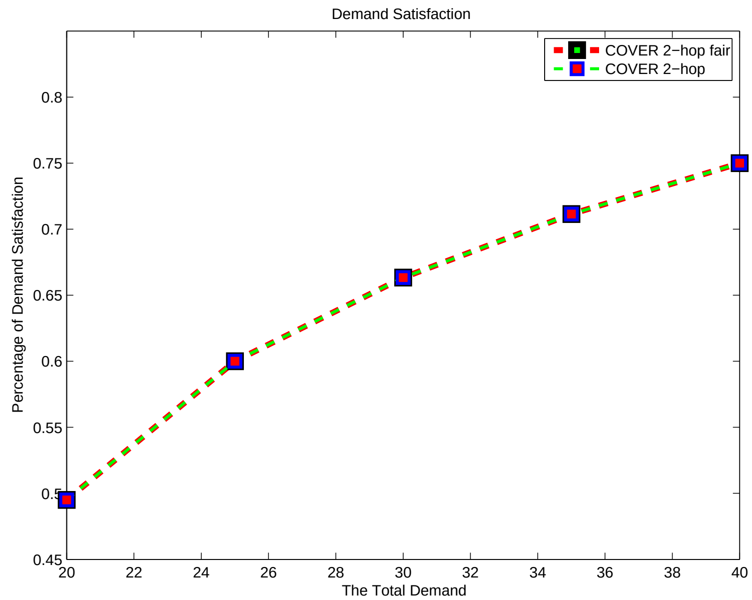


Figure 4.44: The level of demand satisfaction for Fairness-aware Two-hop COVER. The number of robots=demand-10, area size=150m x 150m, communication range=50m

fact that each robot considers only the highest demanding landmark to associate with it not the closest one. So, if there are two landmarks and one is close to it with a small demand and one is far with a high demand, the robot will choose the farthest one to associate with it, which consequently causes the robot to move higher distance. The same holds for the total time as in Figure. 4.46. Finally, the total messages increases by 10-30% in the new approach as in Figure. 4.47. In the new approach, each landmark will start the cooperation earlier than in the previous approach. When a landmark achieves its minimum level of demand satisfaction it will start the cooperation while in the previous approach it will only cooperate when its demand is zero.

We also show the effect of increasing the number of robots for the same level of demand in the Fairness-aware Two-hop COVER. We see in Figure. 4.48 that

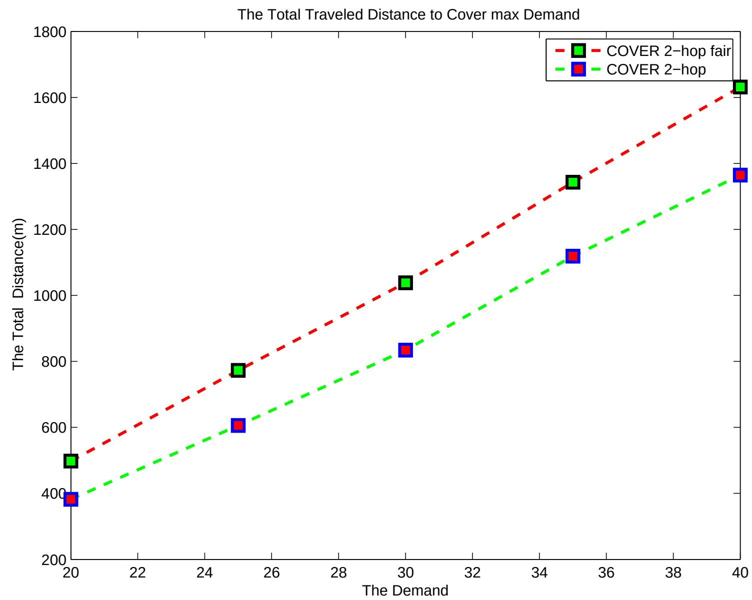


Figure 4.45: The total distance traveled to achieve the maximum possible demand satisfaction in Fairness-aware Two-hop COVER. The number of robots=demand-10, area size=150m x 150m, communication range=50m

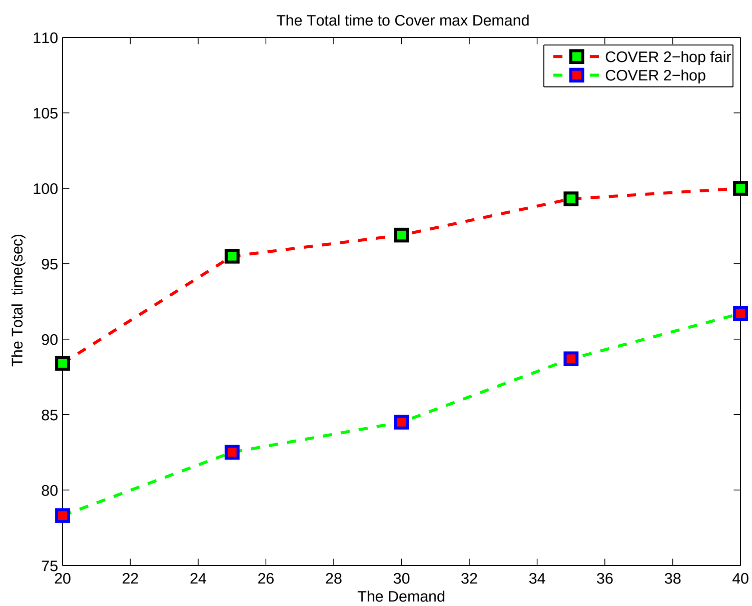


Figure 4.46: The total time traveled to achieve the maximum possible demand satisfaction in Fairness-aware Two-hop COVER. The number of robots=demand-10, area size=150m x 150m, communication range=50m

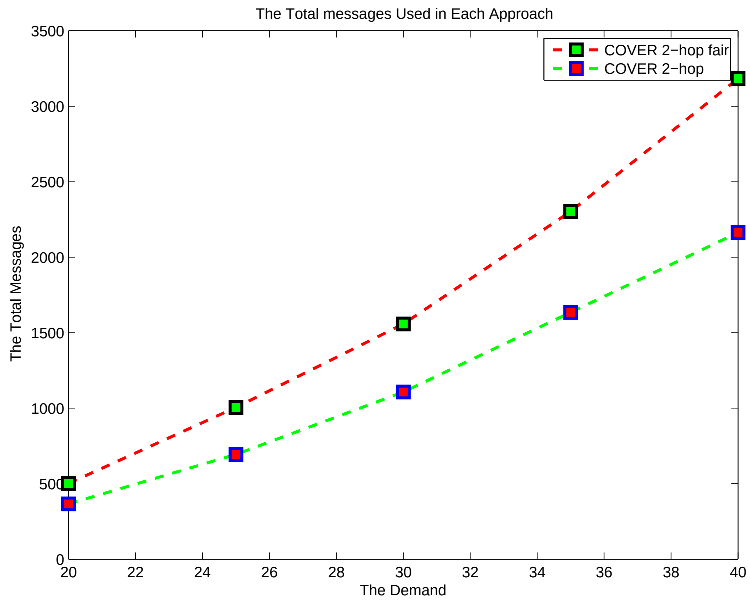


Figure 4.47: The total number of messages in Fairness-aware Two-hop COVER. The number of robots=demand-10, area size=150m x 150m, communication range=50m

increasing the number of robots reduces the standard deviation of the remaining demand until it reaches zero when all the demand is satisfied.

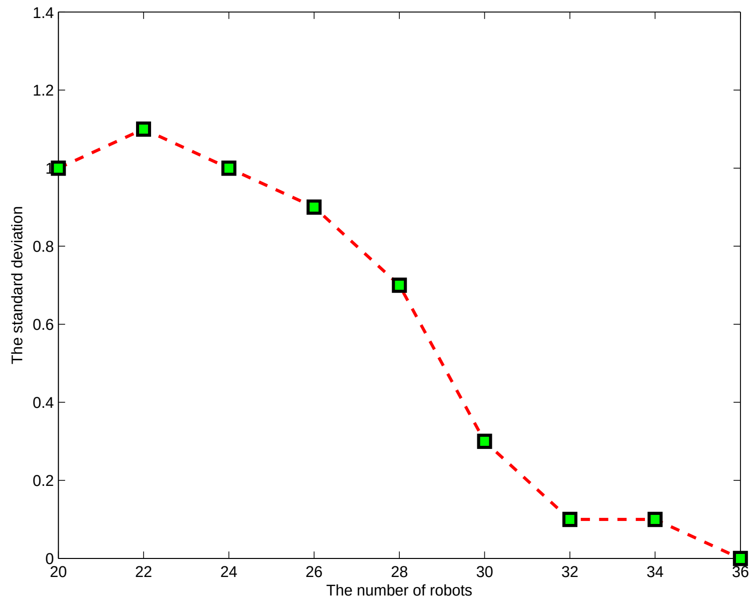


Figure 4.48: The standard deviation in the Fairness-aware Two-hop COVER. The demand=30, area size=150m x 150m, communication range=50m

4.7 Conclusion

In this chapter, we have proposed a novel cooperative and distributive method for multi-robot deployment using virtual force based on landmarks' demand called COVER. COVER is a cooperative approach such that landmarks and robots will help each other to maximize and expedite the demand satisfaction of the landmarks. Moreover, in order to shorten the time and reduce the total distance and messages with also improving the level of demand satisfaction, we have proposed Two-hop COVER that relies on two-hop communication. Finally, since COVER and the Two-hop COVER are not able to always reach 100% demand satisfaction, we proposed a Trace Fingerprint to do so in an efficient way. At the end, we consider the fairness in distributing robots among landmarks in case of the total demand of the landmark is greater than the available robots.

CHAPTER 5

EXPERIMENTAL STUDY

5.1 Introduction

Multi-robots deployment in an unknown environment is a new challenging problem that needs to be investigated to understand the behavior of robotics applications under real parameters. Virtual force has been proposed to tackle such a problem, however, most of the proposed studies did not consider the physical properties of the robots and obstacles and their overall effect on the deployment. In this study, we will consider different realistic parameters. First, we consider the localization of the robots. In a harsh environment, the only convenient way is to use the odometer properties of the robot to induce its location. This approach has its error that accumulates with time. Some of the studies assume that localization is already achieved using GPS [19], however, GPS is not available in indoor places or in the areas that are covered by trees. Additionally, GPS has an error that cannot be tolerated in small-scale scenarios. Another work [20] assumes that a localization method is available and so they build their work based on that assumption, however in a real implementation, localization is a major issue and its error is sometimes very high that affects the overall performance. Another issue is the obstacles presence which can interfere with robots movement during the deployment. Dealing with obstacles is not as simple as assumed by some studies. The presence of obstacles means a high processing of the feedback coming from the distance sensors as well as increasing the path length to the goal. Many previous works have treated obstacles as a repulsive force [19] [1] [21], however in our scenario, we have a goal that the robot needs to go to (which is the landmark) so the robot may need to move around the obstacle to get back again to the path of the goal. More importantly, even if there are no obstacles in the area of interest, the robots themselves could become obstacles for each other.

5.2 A Study of Basic Virtual Force and Full Virtual Force using Webots Simulation

5.2.1 Introduction

In this section, we opt to implement two variants of virtual force based deployment. The first is the full virtual force where robots deployment takes in consideration the robots around and the landmark in virtual force calculation. The robots start to satisfy landmark needs after finishing the deployment. The second approach is the modified virtual force where robots start the deployment and in parallel search for landmarks. Once a robot finds a landmark, it starts to satisfy its demand. The implementation considers a realistic scenario using Khepera III robots on Webots. Webots tool models all the physical properties of Khepera III as well as the obstacles. The controller of each robot incorporates the PID controller along with virtual force to control the robot and avoid obstacles. For localization, we use the odometer properties of the robot to compute its current location given that the initial location is known.

5.2.2 PID Implementation

A proportional integral derivative controller (PID controller) is a general control feedback system used widely in system control [40]. PID algorithm consists of three basic coefficients; Proportional, Integral and Derivative which are varied to get optimal response. The basic idea behind a PID controller is to read a sensor, then compute the desired actuator output by calculating Proportional, Integral, and Derivative responses and summing those three components to compute the output. PID controller and virtual force will be used to control the movement of each robot. In virtual force each robot will communicate with its neighbor robots and based on these communications it will decide its next position. The PID controller will be used to allow a robot to move to its next position and avoid obstacles in its way. In order to make a robot move from one point to another, we use a go-to-goal controller. However, the robot may find obstacles in its way, so we use avoid-obstacles controller. To avoid hard switching between go-to-goal and avoid-obstacles controller and at the same time to make the robot move toward the goal, we use go-to-goal-and-avoid-obstacles controller; the implementation of the three controllers are described below based on the details in reference [41].

1. Go-to-goal controller: in this controller the robot has a goal defined by x and y coordinate. So the input to the controller will be v and w (the linear velocity and angle), and the output will be v and w (the linear and angular velocity). The robot will move according to the output of the controller. Using the feedback from wheels encoders, the controller will adjust v and w continuously until the robot reaches its goal.

2. Avoid-obstacles controller: The objective of this controller is to steer the robot away from obstacle to avoid a collision. This is known as the avoid-obstacles behavior. The IR sensors allow to measure the distance to obstacles in the environment, but we need to compute the points in the world to which these distances correspond. So, each robot will get IR sensors reading, Khepera III has nine IR sensors. The IR sensors reading will be mapped to a distance. Based on these calculations, the presence of an obstacle and its position is determined. Then the controller uses these readings to compute v and w to steer the robot away from the obstacle. This process is repeated until the IR readings are below a threshold which makes the robot either to switch to the go-to-goal controller or go-to-goal-and-avoid-obstacles controller.

3. Go-to-goal-and-avoid-obstacles controller: in this controller, there is one vector pointing to the goal from the robot and another vector pointing from the robot to a point in the space away from the obstacle. These two vectors are combined (blended) into a component vector, which is a vector that points the robot both away from the obstacle and towards the goal

5.2.3 Performance Evaluation

In this work, two algorithms are implemented for robots self-deployment in unknown environment: Full virtual force, and Modified virtual force. These algorithms have been implemented in our previous work [39].In addition, each robot runs the PID controller in order to consider all the physical properties of the robots as well as a real environment. For localization, we use the odometer properties of the robot to compute its coordinates given that the initial location is known. Since we consider a small-scale scenario in this work, the error of this localization approach can be tolerated.

5.2.3.1 Full Virtual Force

In this approach, the robots will implement the virtual force until they reach the equilibrium state. If a robot receives any demand message from landmarks during the deployment phase, it will only consider its demand in virtual force calculations. Only the landmarks that have a demand can exert the attractive force, otherwise zero force is exerted. The force is set relatively to the demand needed by the landmark; the higher the demand, the higher the attractive force.

After finishing the deployment, the robots will start listening for the demand messages and applying the association algorithm to get associated with one of the neighboring landmarks.

5.2.3.2 Modified Virtual Force

In this approach, the robots will use the original virtual force to deploy themselves in the area of interest. Once any of the robots hears a demand message from a landmark, it will start the association process. If it succeeds, it will move toward that landmark, if not it will continue moving using virtual force.

5.2.3.3 Simulation Setup

The simulation has been conducted using Webots tool. It is a high fidelity simulator that has a modeling for many realistic robots. All the physical characteristics of the robots and the environment are represented. We opt to use Khepera III robot. The Khepera III robot integration is a product of the Swiss company K-Team. Features available on the platform include multiple sensor arrays for both long range and short range object detection, swappable battery pack system for optimal autonomy, and differential drive odometry [42]. The Khepera III robot embeds lots of sensors and advanced computational power which enhances its application in education and research. Khepera III is equipped with 11 infrared (IR) range sensors, of which nine are located in a ring around it and two are located on the underside of the robot as shown in Figure. 5.1. The IR sensors are complemented by a set of five ultrasonic sensors. Webots simulator has an interesting feature that the implemented code on Khepera robot on simulation can easily be transferred to a real robot without a modification.

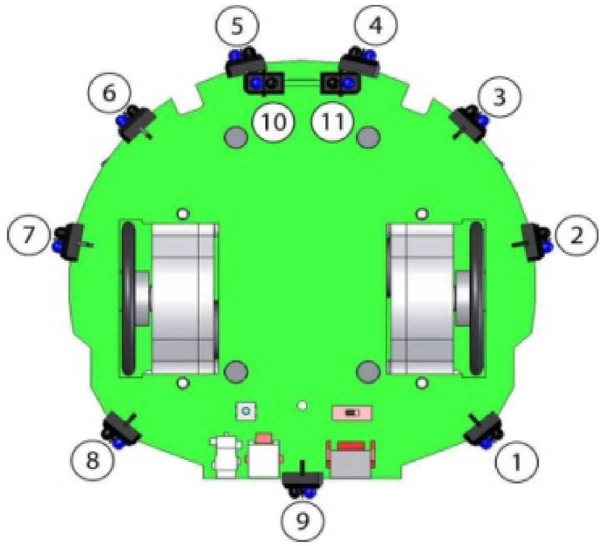


Figure 5.1: The top view of IR sensors in Khepera III

Simulation parameters are shown in Table 5.1. Each experiment is repeated 15 times with each experiment the position of the robots, landmarks, obstacles, and the demand distribution are set randomly. The performance metrics used in this study are:

1. Demand satisfaction: this metric measures the percentage of demand that is satisfied by the end of the implementation of the algorithm.

2. Total traveled distance.

3. Total time.

4. Total messages: this metric counts the number of messages that are utilized in the implementation of the two algorithms.

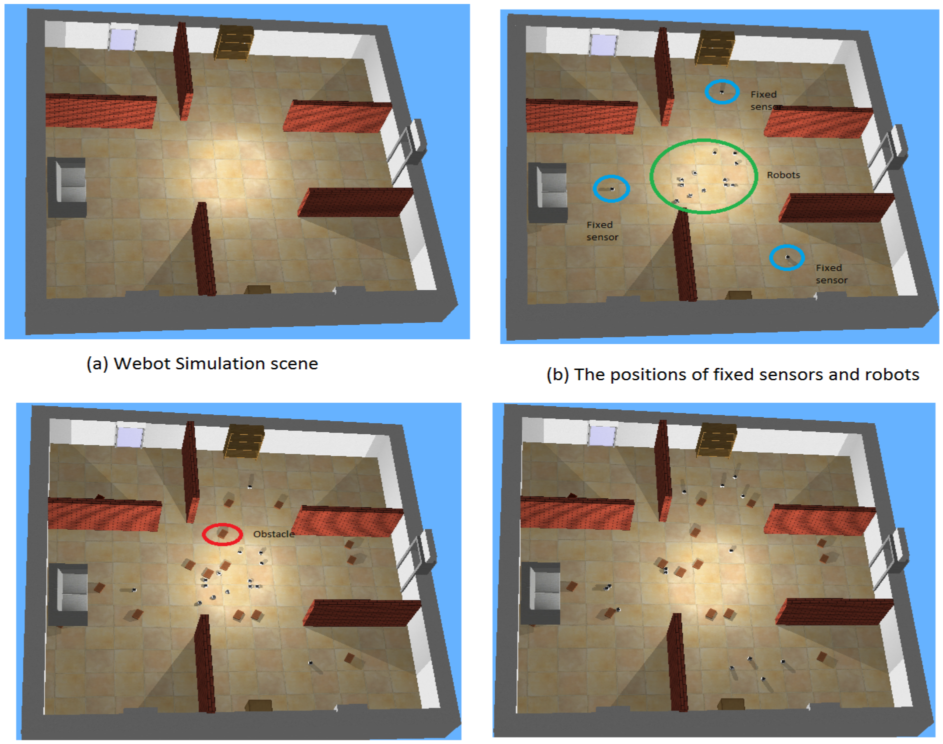
All the above metrics can be used to implicitly measure the energy consumption because the total distance and messages are the main sources of energy consumption. The simulation area is 10m x 10m as shown in Figure. 5.2. Obstacles are small rectangles of size 30cm x 20cm distributed randomly throughout the area.

Parameters	**Value**
Simulation tool	Webot 8.1.0
Robot name	Khepera III
Number of robots (randomly distributed)	15, 20, 25
Number of Landmarks (randomly distributed)	6
Demand (robots)	15
Area size	10m x 10m
Number of obstacles (randomly distributed)	20, 40
Obstacle size	0.3m x 0.2m
Number of experiments	15

Table 5.1: Webots Simulation Parameters

5.2.3.4 Results and Analysis

The performance of the two algorithms has been investigated in different setups to explore the effect of using real robots in real environments. Additionally, the effect of increasing the number of obstacles is studied as well. First we show the effect of obstacles on modified virtual force algorithm. We can observe that the demand satisfaction is affected by the presence of obstacles and their density as in Figure. 5.3, but the main effect of obstacles is on the other metrics: average distance, total time, and messages as shown in Figure. 5.4, 5.5, 5.6. The presence



(a) Webot Simulation scene

(b) The positions of fixed sensors and robots

(c) The presence of obstacles

(d) The final deployment

Figure 5.2: The deployment process. A) The structure of the area of interest. B) Three landmarks with blue circles and 12 robots inside the green circle. C) Obstacles present, one of them is inside a red circle. D) The final deployment of the robots.

of obstacles causes the robot to move longer distance searching for landmarks. Consequently, the total time needed to achieve the same percentage of demand satisfaction increases with the increase of obstacles density as in Figure. 5.4. For the total messages, the obstacles delay robots from getting associated to a landmark which causes more messages to be exchanged in order to implement the algorithm. We conclude that ignoring a real factors such as obstacles can lead to performance statistics that are far from real and consequently misleading conclusions.

We also compare full virtual force and modified virtual force to show the effect

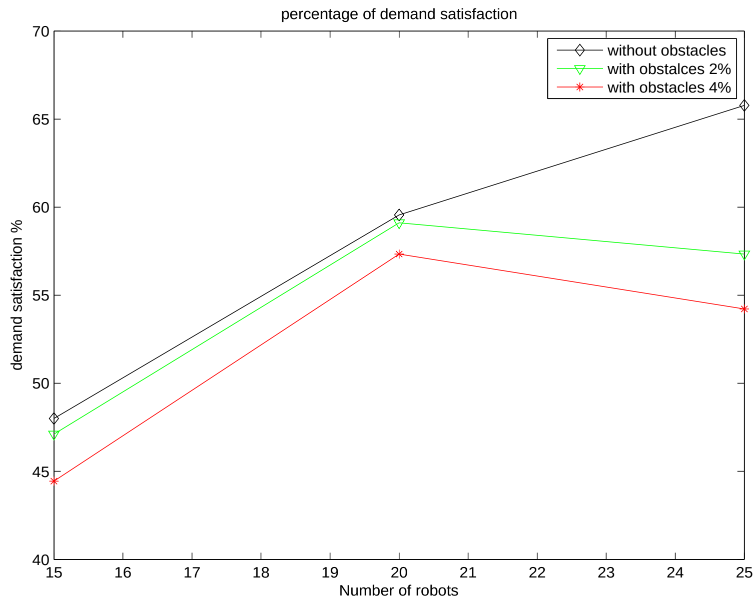


Figure 5.3: Percentage of demand satisfaction using modified virtual force algorithm

of obstacles on both of them. While the performance of both of them is close in term of demand satisfaction as in Figure. 5.7, we see that modified virtual force is better in term of average distance, total time, and total messages as in Figure. 5.8, 5.9, 5.10.

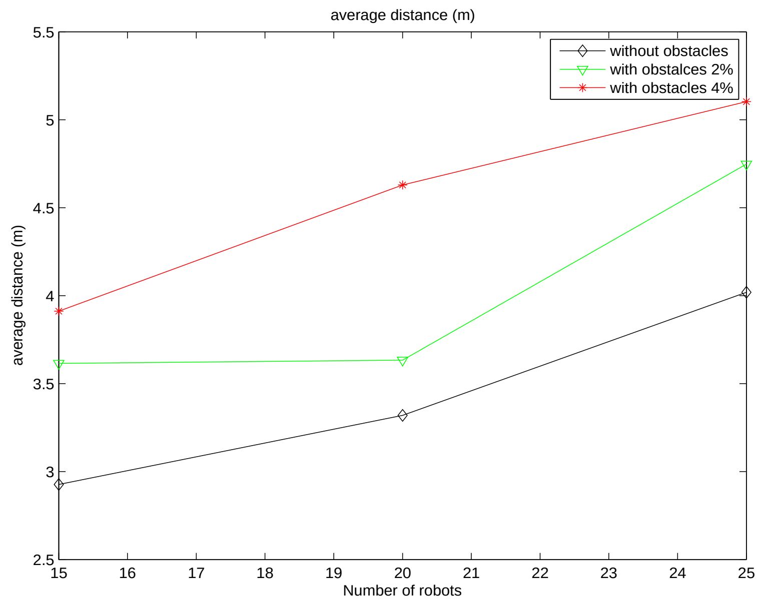


Figure 5.4: The average traveled distance of each robot to satisfy the given demand using modified virtual force algorithm

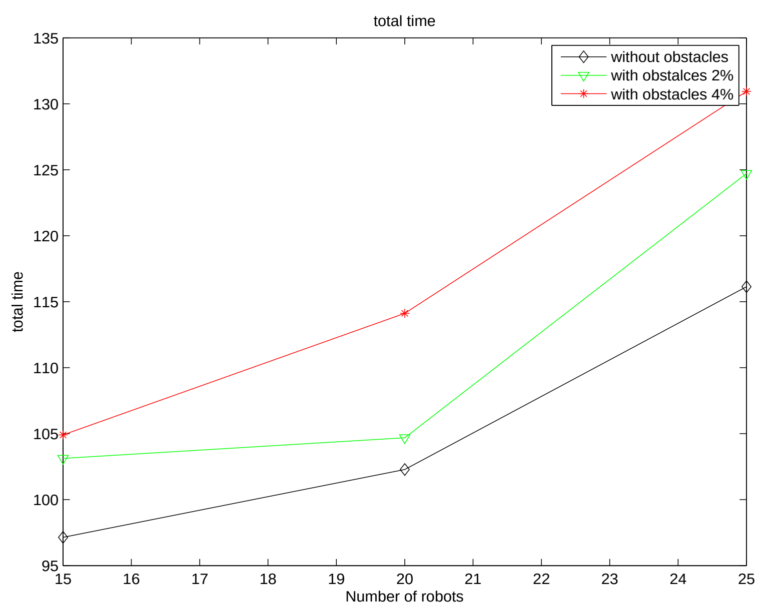


Figure 5.5: The total time needed for each robot to satisfy the given demand using modified virtual force algorithm

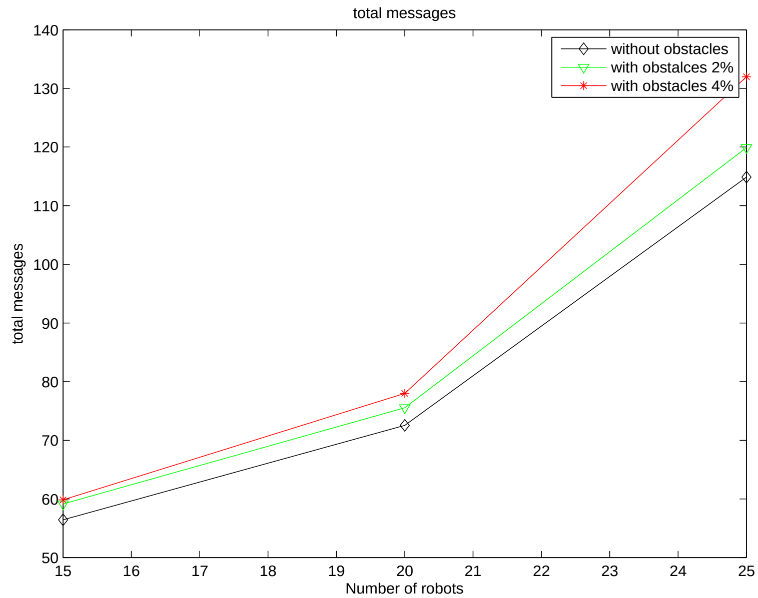


Figure 5.6: The total messages utilized by each robot to satisfy the given demand using modified virtual force algorithm

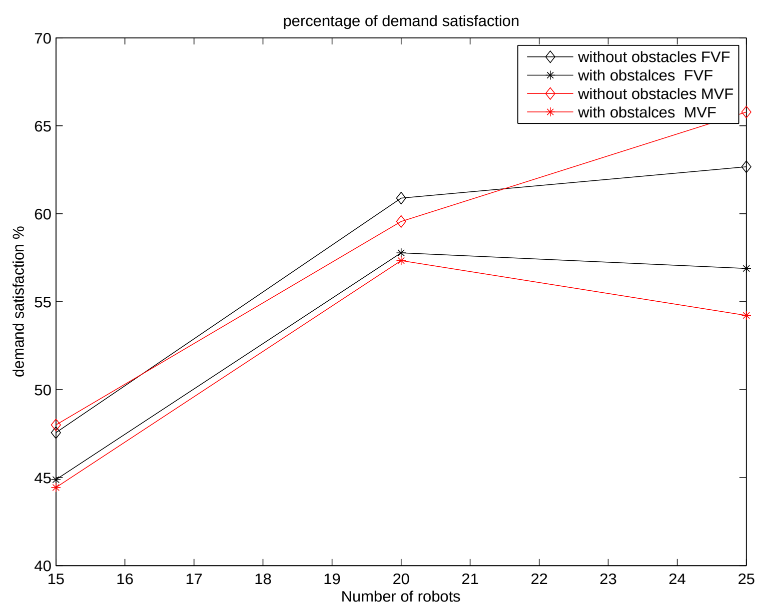


Figure 5.7: Percentage of demand satisfaction using modified virtual force and full virtual force algorithms

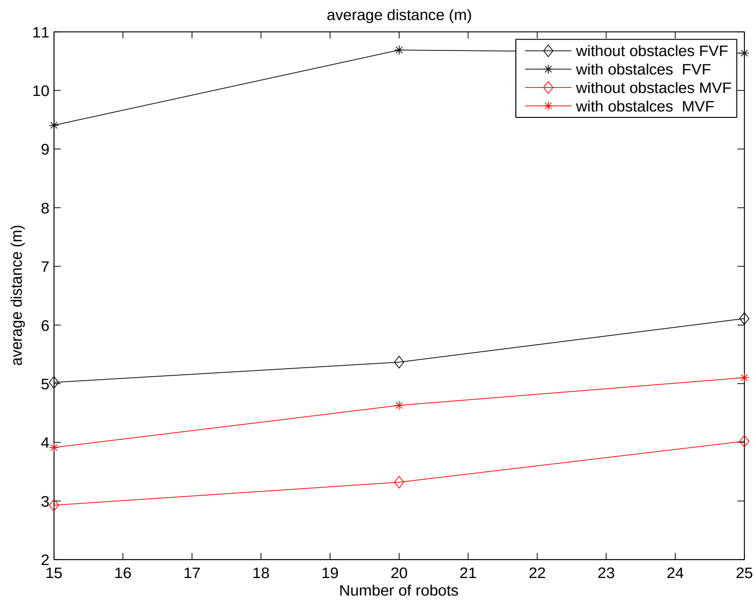


Figure 5.8: The average traveled distance of each robot to satisfy the given demand using modified virtual force and full virtual force algorithms

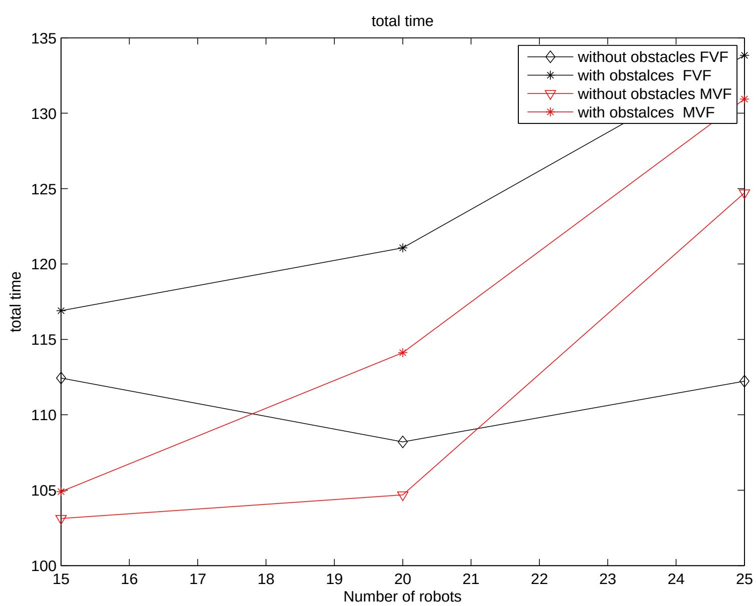


Figure 5.9: The total time needed for each robot to satisfy the given demand using modified virtual force and full virtual force algorithms

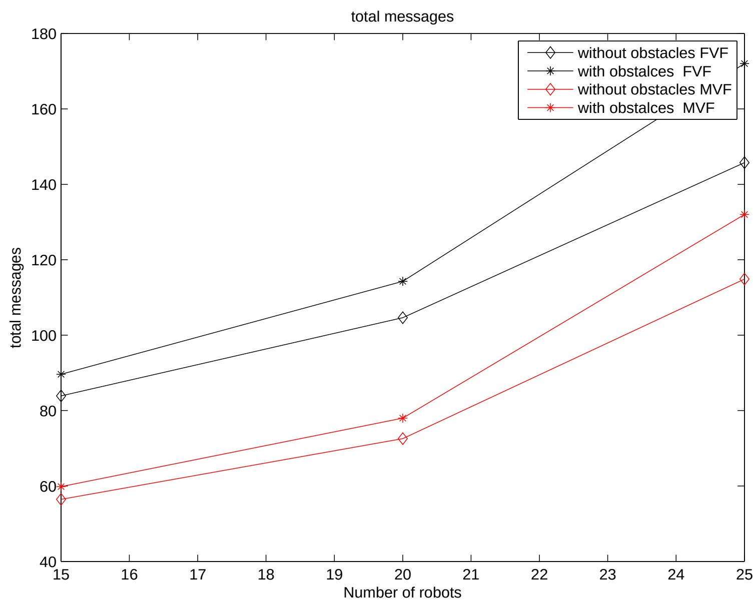


Figure 5.10: The total messages utilized by each robot to satisfy the given demand using modified virtual force and full virtual force algorithms

5.3 A Study of COVER Using TurtleBot

In this section, we are going to show some real experimental results on Turtlebot2 robots to provide a proof of concept of COVER algorithm. TurtleBot is a low-cost, personal robot kit with open source software. It consists of a mobile base, laptop computer, and 3D depth camera. The camera is used for navigation and obstacle avoidance given the boundary of the area. In order to conduct our experiments, we use four robots and two laptops as landmarks. The experiments are conducted on area of 20m * 20m.

5.3.1 Basic Virtual Force Using TurtleBot

First we implemented the original virtual force using four Turtlebot robots. The robots are placed at the center of the area and the goal is to spread them over the area and keep the distance between each neighbors at 4 meters. The robots are put at the center as in Figure. 5.12-A then as a result of virtual force effect, the robots move multiple steps until they reach the equilibrium state as in figure 5.12-C. For path planing and obstacle avoidance, each robot uses the 3D depth camera placed in front of it to decide the distance to obstacles. Each robot will need to know the boundary of the area and its initial position within the area.

5.3.2 COVER Using Turtlebot

We have also implemented the proposed deployment approach (COVER) using Turtlebot robots. The experimental parameters are shown in Table 5.2. Different

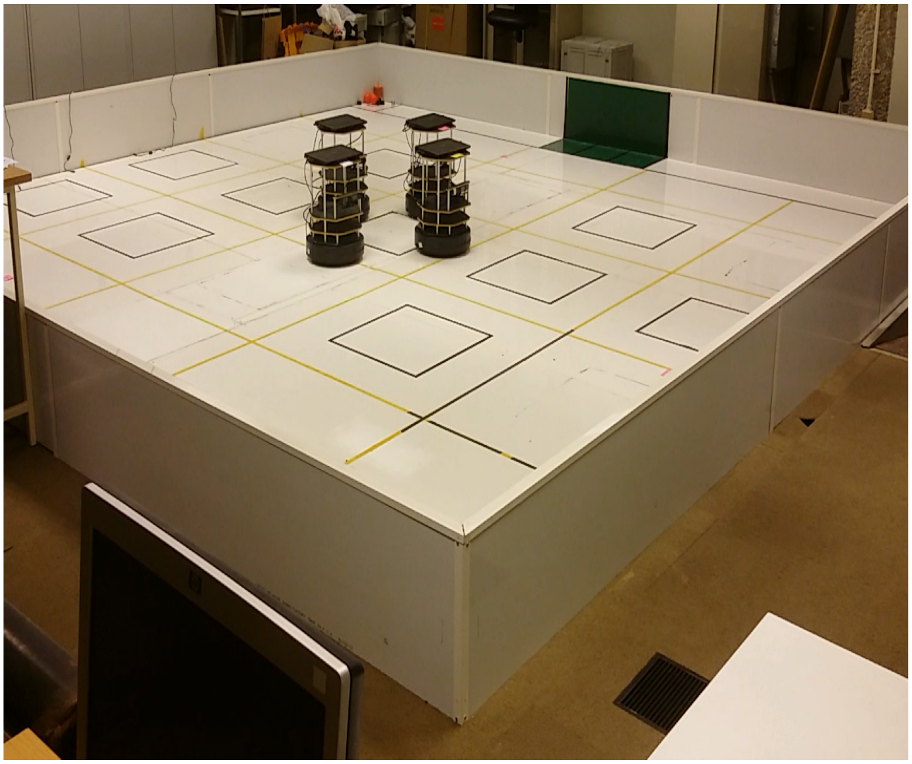
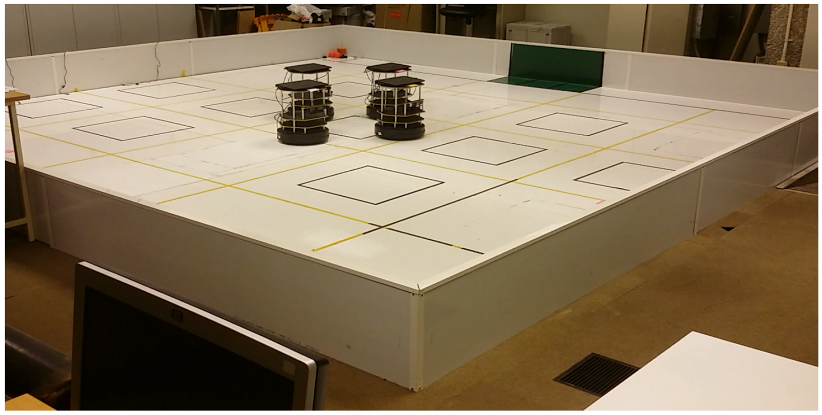
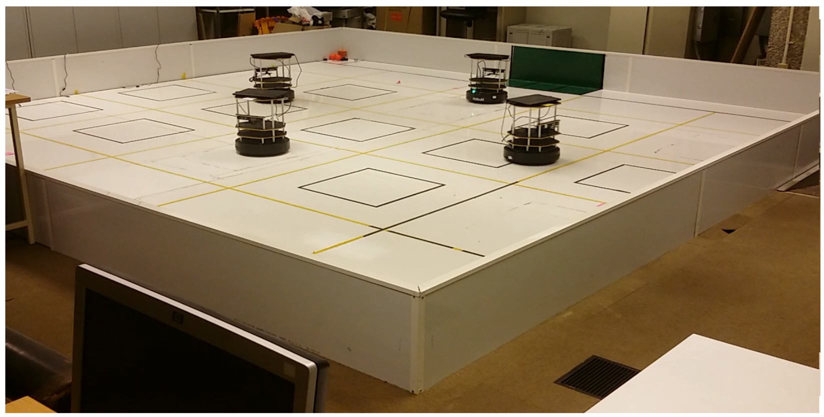


Figure 5.11: An example of the experimental area where four robots are placed at the center.

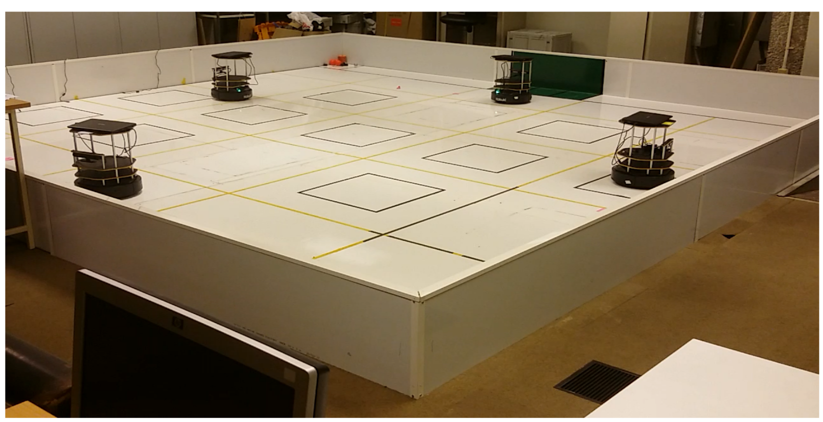
experiments are conducted utilizing different combinations of robots and landmarks and the demand. The first experiment is conducted using 4 robots and two landmarks. The robots are initially at the center of the area and the landmarks one at the right most point and one at the left most point as shown in 5.13-A. The robots will initially move according to virtual force in order to increase the chances of locating a landmark. The robots position will be as in figure 5.13-B . Then robot R_3 will hear a demand from landmark L_1 and associate to it. In the same time robot R_4 will hear a demand from landmark L_2 and associate to it. Since the demand of the landmark L_2 is still not satisfied, robot R_4 will attract robot R_1 and robot R_2 until they become in the range of landmark L_2 and associate to it as in Figure. 5.13-C.



(A)



(B)



(C)

Figure 5.12: An example of virtual force implementation using turtlebot robot. A) The initial position of the four robots at the center of the area. B) The position of the robots after sometime of implementing virtual force. C) The final position of the robots where they reached the equilibrium and the distance between each pair is around 4 meters

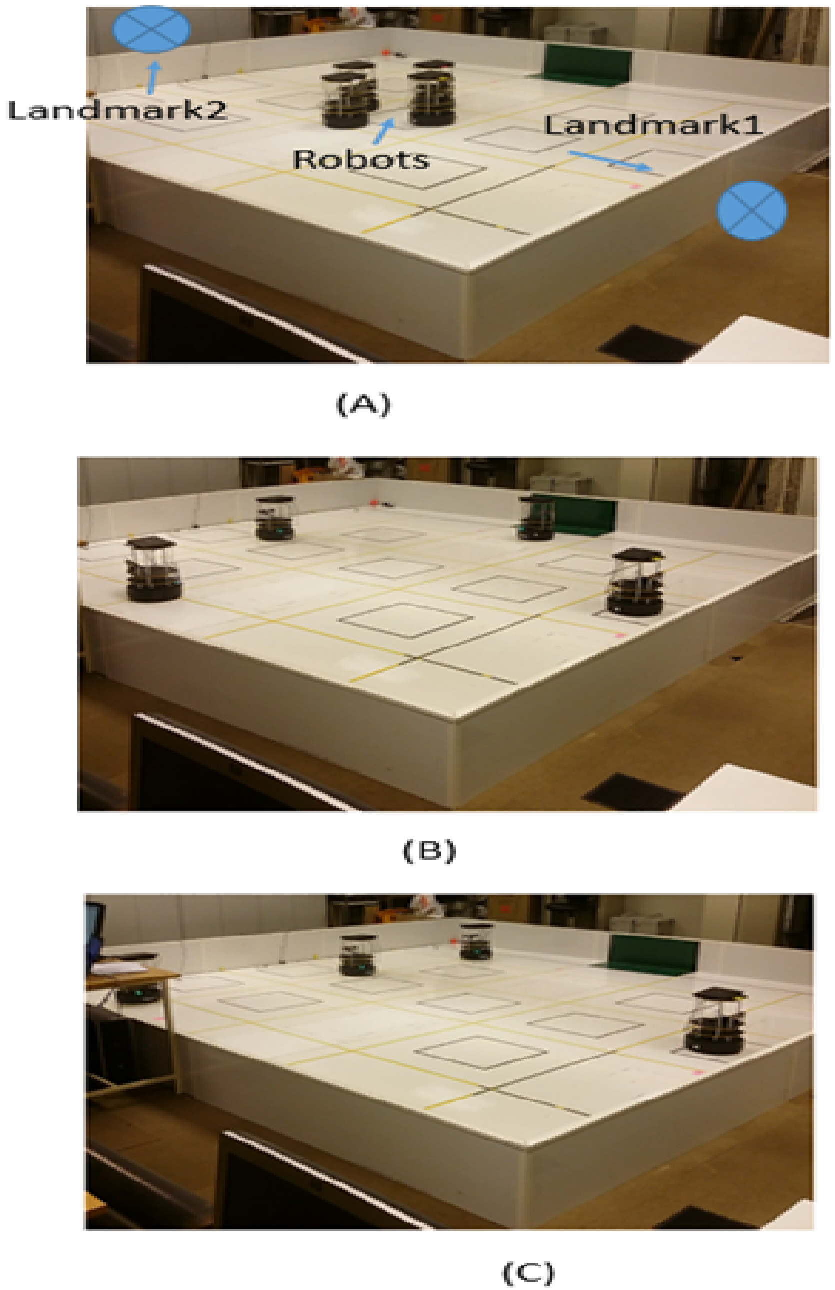
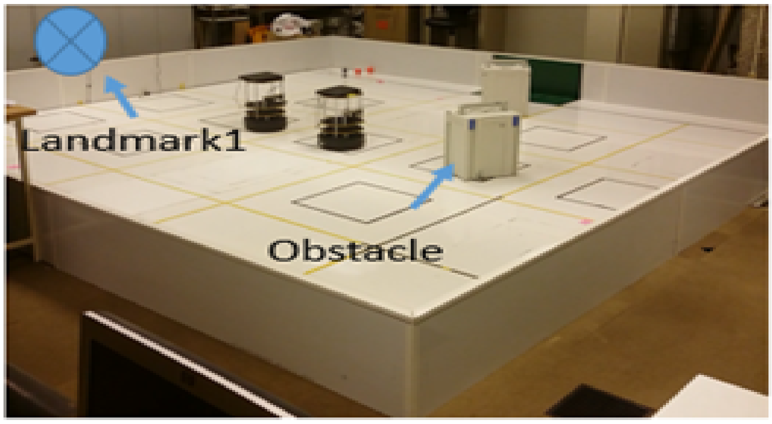


Figure 5.13: Robots deployment according to COVER approach. A) Four robots are placed at the center and the landmarks one at (20, 3) with demand of 1 and one at (0, 3) with demand of 3. B) Robot R_3 and R_4 were able to locate landmark L_1 and L_2 respectively and associate to them. C) Robot R_4 attracts robot R_1 and R_2 until they become in the range of landmark L_2 to associate with it

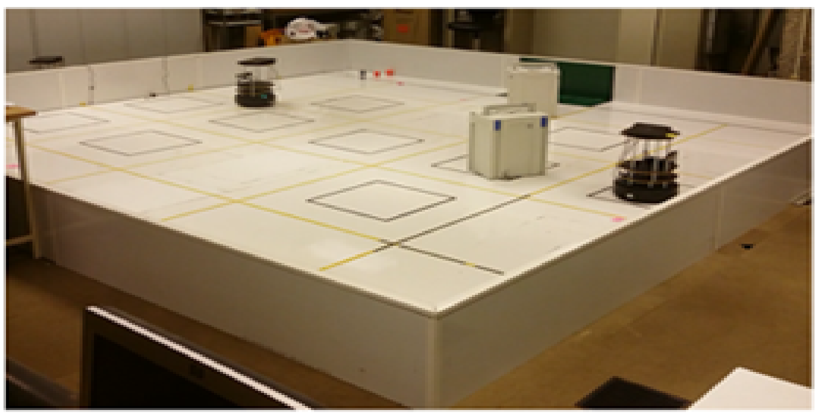
Parameters	Value
Robot Name	TurtleBot
Number of robots	4
Robot's transmission range	8m
Landmark's transmission range	8m
Number of Landmarks	1, 2
Demand (robots)	1, 4
Area size	20m x 20m
Number of obstacles	1
Obstacle size	0.5m x 0.7m

Table 5.2: Experiment Parameters

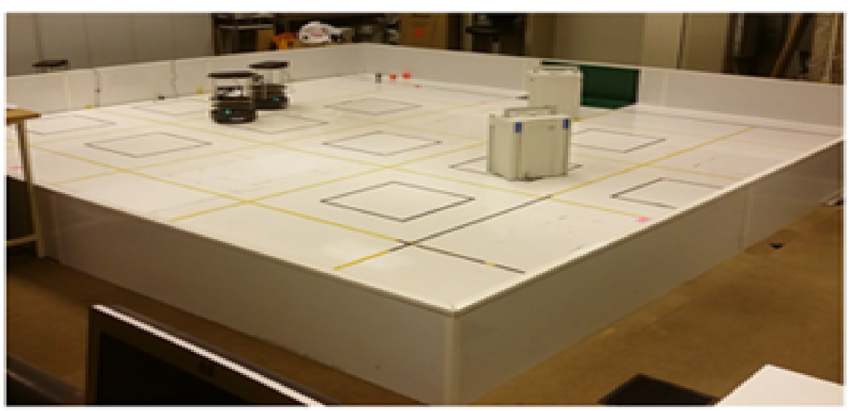
In order to show an obstacle avoidance scenario, we placed two obstacles in the area of interest and two robots as in Figure. 5.14-A. The two robots will move according to virtual force. Once one of the robots finds the landmark with demand it will attract the other robots toward it. In each movement of robot R_1 it will avoid the obstacle in its way and reach the goal successfully. First robot R_2 will move according to virtual force to the position (12, 3). In its path to the goal there is an obstacle in its way and using the 3D depth camera equipped with the robot, it will recognize that there is an obstacle and avoid it as in figure 5.14-B. Then robot R_1 will attract robot R_2 toward landmark L_1 and the path that robot R_2 will move through it is having an obstacle so the robot will plan its path accordingly as in figure 5.14-C.



(A)



(B)



(C)

Figure 5.14: Robots deployment according to COVER approach in the presence of obstacle. A) The initial position of the robots. B) Robots after virtual force calculations and robot R_1 gets associated. C) Robot R_1 attracts robot R_2 toward landmark L_1

CHAPTER 6

CONCLUSION AND FUTURE DIRECTIONS

In this thesis, we have studied virtual force approach in order to come up with a framework that considers different aspects of robot deployment using virtual force. First, we study the main parameters that virtual force relies on which are the attractive force w_a and the repulsive force w_r. We present two calibrates for each one in order to adapt to different scenarios. Considering the remaining energy of each robot is also factored in virtual force calculation to balance energy consumption among robots. In order to utilize virtual force for purposeful deployment, we proposed a cooperative virtual force technique called COVER. This approach modifies the original virtual force such that the demand of landmarks is factored during virtual force implementation. This approach is shown to have some limitations such as robot deadlock or increase in the distance traveled. So, to address the aforementioned problems we proposed Two-hop COVER that utilizes two-hop communications between robots and landmarks to improve the performance of the deployment. Moreover, in order to guarantee 100% demand satisfaction of the landmarks, we proposed Trace Fingerprint method that should be used with Two-hop COVER to guarantee the maximum possible demand satisfaction. Finally, the fairness in distributing robots among landmarks is considered in the scenarios when the collective demand of the landmarks is greater than the available robots. All the above approaches have been validated through extensive simulation experiments. In addition, a proof-of-concept experiment using Turtlebot robots has been carried out to see a real implementation of COVER. Also, Khepera III robots available on Webots have been used to assess the original virtual force and full virtual force approach. As for future improvements, the following can be used as a guidance for future improvements.

1. Propose an Adaptive Virtual Force. One of the main challenges is how to make virtual force adapt itself to perform the required deployment online. In the proposed setting, we assume that some details will be fed to the system such as the number of robots, then we set the attractive force and the repulsive force accordingly. But, what if the robots detected any changes in the environment and figured out that the deployment is not reaching its goal, in this case, it would be better for w_a and w_r to be changing autonomously to adapt to the environmental changes and to improve the deployment.

2. Advance Dealing with Obstacles. In the current implementation of virtual force in the literature, obstacles are treated as an object that will exert a

repulsive force, however, dealing with the size of obstacles and how obstacles will be detected worth further investigations. For instance, a robot can detect obstacle using the distance sensors as in Figure 6.1, then the robot starts to make a conception about the shape of the obstacle. Based on the built conception, the repulsive force should be exerted. Additionally, we would need to look on how to deal with regular and irregular obstacles and their effects on the stabilization of virtual force based deployment.

3. Real experiments of the other proposed algorithms (Two-hop COVER and Trace Fingerprint) would give a great insight of their performance in a real environment.

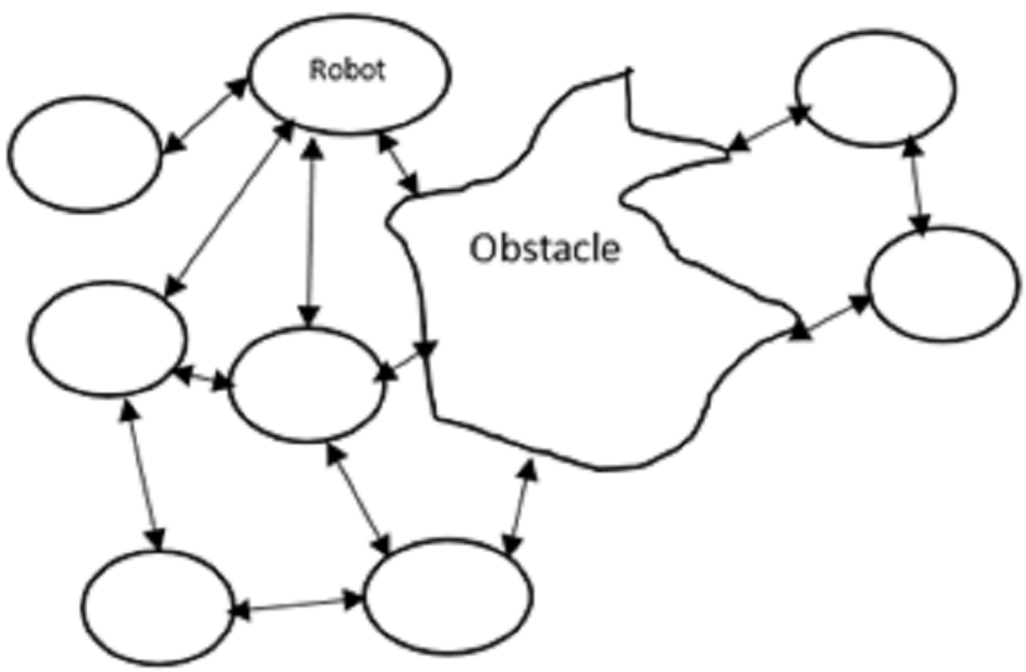


Figure 6.1: An example of the interaction between robots and obstacles

REFERENCES

[1] Y. Zou and K. Chakrabarty, "Sensor Deployment and Target Localization Based on Virtual Forces," in *Twenty-Second Annual Joint Conference of the IEEE Computer and Communications (INFOCOM)*, vol. 2, no. C, 2003, pp. 1293 – 1303.

[2] G. Tan, S. A. Jarvis, and A.-M. Kermarrec, "Connectivity-Guaranteed and Obstacle-Adaptive Deployment Schemes for Mobile Sensor Networks," *Distributed Computing Systems, 2008. ICDCS '08. The 28th International Conference on*, vol. 8, pp. 429–437, 2008. [Online]. Available: papers2://publication/uuid/B4FE933E-1EDB-41A1-AE5D-EB7FE2F27999

[3] C. Costanzo, V. Loscrí, E. Natalizio, and T. Razafindralambo, "Nodes self-deployment for coverage maximization in mobile robot networks using an evolving neural network," *Computer Communications*, vol. 35, no. 9, pp. 1047–1055, 2012. [Online]. Available: http://dx.doi.org/10.1016/j.comcom.2011.09.004

[4] M. Gupta, C. R. Krishna, D. Prasad, and I. Scholar, "SEEDS : Scalable

Energy Efficient Deployment Scheme for Homogeneous Wireless Sensor Network," in *the International Conference on Issues and Challenges in Intelligent Computing Techniques (ICICT)*, 2014.

[5] M. Erdelj, T. Razafindralambo, and D. Simplot-Ryl, "Covering points of interest with mobile sensors," *IEEE Transactions on Parallel and Distributed Systems*, vol. 24, no. 1, pp. 32–43, 2013.

[6] X. Li, H. Frey, N. Santoro, and I. Stojmenovic, "Strictly localized sensor self-deployment for optimal focused coverage," *IEEE Transactions on Mobile Computing*, vol. 10, no. 11, pp. 1520–1533, 2011.

[7] D. Zorbas and T. Razafindralambo, "Wireless Sensor Network Redeployment under the Target Coverage Constraint," in *New Technologies, Mobility and Security (NTMS), 2012 5th International Conference on*, 2012, pp. 1–5.

[8] M. R. Senouci, A. Mellouk, K. Assnoune, and F. Bouhidel, "Movement-assisted Sensor Deployment Algorithms: a Survey and Taxonomy," *IEEE Communications Surveys & Tutorials*, vol. 17, no. 4, pp. 2493 – 2510, 2015. [Online]. Available: http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=7052338

[9] G. Wang, G. Cao, and T. L. Porta, "Movement-assisted sensor deployment," *Proceedings - IEEE INFOCOM*, vol. 4, no. 6, pp. 2469–2479, 2004.

[10] G. Wang, G. Cao, P. Berman, and T. F. La Porta, "Bidding protocols for deploying mobile sensors," *IEEE Transactions on Mobile Computing*, vol. 6, no. 5, pp. 515–528, 2007.

[11] M. R. Senouci, A. Mellouk, and K. Assnoune, "Localized Movement-Assisted SensorDeployment Algorithm for HoleDetection and Healing," *IEEE Transactions on Parallel and Distributed Systems*, vol. 25, no. 5, pp. 1267–1277, 2014. [Online]. Available: http://ieeexplore.ieee.org/articleDetails.jsp?arnumber=6519242

[12] B. J. Farahani, H. Ghaffarian, and M. Fathy, "A Fuzzy Based Priority Approach in Mobile Sensor Network Coverage," *International Journal of Recent Trends in Engineering*, vol. 2, no. 1, 2009.

[13] H. Shu and Q. Liang, "Fuzzy Optimization for Distributed Sensor Deployment," *Communications Society*, pp. 1903–1908, 2005.

[14] X. Yu, "A faster convergence artificial bee colony algorithm in sensor deployment for wireless sensor networks," *International Journal of Distributed Sensor Networks*.

[15] Y. Suen, "A Genetic-Algorithm Based Mobile Sensor Network Deployment Algorithm ," *Journal of Chemical Information and Modeling*, vol. 53, no. 9, pp. 1689–1699, 2013.

[16] X. H. Wei Wang, "Research on Sensor Network Self-deployment with Virtual Attractive and Repulsive Forces," *INTERNATIONAL*

JOURNAL ON Advances in Information Sciences and Service Sciences, vol. 5, no. 6, pp. 1031–1037, Mar. 2013. [Online]. Available: http://www.aicit.org/aiss/global/paper_detail.html?jname=AISS&q=2816

[17] A. Howard, M. J. Mataric, and G. S. Sukhatme, "Mobile sensor network deployment using potential fields: A distributed, scalable solution to the area coverage problem," in *Proceedings of the 6th International Symposium on Distributed Autonomous Robotics Systems (DARS02)*, vol. 5, 2002, pp. 299–308. [Online]. Available: http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.17.461&rep=rep1&type=

[18] E. Gelenbe, S. Timotheou, and D. Nicholson, "Fast distributed near-optimum assignment of assets to tasks," *The Computer Journal*, vol. 53, no. 9, pp. 1360–1369, 2010.

[19] J. Chen and X. Wang, "The Coverage Problem in Heterogeneous Wireless Sensor Network: An Improved Algorithm of Virtual Forces," *Proceedings of 2013 Chinese Intelligent Automation Conference*, vol. 255, no. 69, pp. 619–630, 2013. [Online]. Available: http://link.springer.com/10.1007/978-3-642-38460-8

[20] N. Ahmed, S. S. Kanhere, and S. Jha, "Ensuring Area Coverage in Hybrid Wireless Sensor Networks," in *Proceedings of the 3rd international conference on Mobile ad-hoc and sensor networks*, 2007, pp. 548–560.

[21] X. Yu, W. Huang, J. Lan, and X. Qian, "A Novel Virtual Force Approach for Node Deployment in Wireless Sensor Network," *2012 IEEE 8th International Conference on Distributed Computing in Sensor Systems*, no. 2011, pp. 359–363, May 2012. [Online]. Available: http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=6227769

[22] H. W. Kuhn, "The Hungarian Method for the assignment problem," *Naval Research Logistics Quarterly*, vol. 2, pp. 83–97, 1955.

[23] M. Garetto, M. Gribaudo, C.-f. Chiasserini, and E. Leonardi, "A Distributed Sensor Relocatlon Scheme for Environmental Control," in *2007 IEEE Internatonal Conference on Mobile Adhoc and Sensor Systems*, 2007, pp. 1–10. [Online]. Available: http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=4428663

[24] Y. Zhang and Z. Wei, "On deployment optimization strategy for hybrid wireless sensor networks," *The 26th Chinese Control and Decision Conference (2014 CCDC)*, pp. 1875–1880, May 2014. [Online]. Available: http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=6852475

[25] J. Roselin and P. Latha, "Energy Balanced Dynamic Deployment Optimization to Enhance Reliable Lifetime of Wireless Sensor Network," *International Journal of Engineering and Technology (IJET)*, vol. 5, no. 4, pp. 3450–3460, 2013.

[26] T. La Porta, “Proxy-based sensor deployment for mobile sensor networks,” *2004 IEEE International Conference on Mobile Ad-hoc and Sensor Systems (IEEE Cat. No.04EX975)*, pp. 493–502, 2004. [Online]. Available: http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=1392190

[27] R.-S. Ko and C.-m. Chen, “Prediction Approaches for Improving Energy Ef fi ciency of Virtual Force Algorithms to the Mobile Sensor Deployment,” in *Proceedings of the 5th ACM international workshop on Mobility management and wireless access*, 2007, pp. 144–147. [Online]. Available: http://dl.acm.org/citation.cfm?id=1298117

[28] T. Tsuchiya and T. Kikuno, “A self-organizing technique for sensor placement in wireless micro-sensor networks,” *18th International Conference on Advanced Information Networking and Applications, 2004. AINA 2004.*, vol. 1, pp. 78–83, 2004. [Online]. Available: http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=1283891

[29] S. Li, C. Xu, W. Pan, and Y. Pan, “Sensor deployment optimization for detecting maneuvering targets,” *2005 7th International Conference on Information Fusion*, p. 7 pp., 2005. [Online]. Available: http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=1592051

[30] N. Guofang, C. Zhongnan, L. Minqiang, H. Liang, and A. Ajith, “Distributed deployment algorithm based on boundary expansion and virtual force for

mobile sensor networks," *Neural Network World*, vol. 24, no. 3, pp. 309–332, Jul. 2014. [Online]. Available: http://www.nnw.cz/obsahy14.html#24.018

[31] J. Chen, S. Li, and Y. Sun, "Novel Deployment Schemes for Mobile Sensor Networks," *Sensors*, vol. 7, no. 11, pp. 2907–2919, Nov. 2007. [Online]. Available: http://www.mdpi.com/1424-8220/7/11/2907/

[32] Q. Li, M. D. Rosa, and D. Rus, "Distributed algorithms for guiding navigation across a sensor network," in *Proceedings of the 9th annual international conference on Mobile computing and networking (mobicom 2003)*, 2003, pp. 313–325.

[33] A. Filippoupolitis and E. Gelenbe, "A distributed decision support system for building evacuation," *Proceedings - 2009 2nd Conference on Human System Interactions, HSI '09*, pp. 323–330, 2009.

[34] G. Gorbil and E. Gelenbe, "Opportunistic communications for emergency support systems," *Procedia Computer Science*, vol. 5, pp. 39–47, 2011. [Online]. Available: http://dx.doi.org/10.1016/j.procs.2011.07.008

[35] G. C. S. Li, A. Zhan, X. Wu, P. Yang, "Efficient Emergency Rescue Navigation with Wireless Sensor Networks," *Journal of Information Science and Engineering*, vol. 64, pp. 51–64, 2011.

[36] A. Ko and H. Y. K. Lau, "Robot Assisted Emergency Search and Rescue System With a Wireless Sensor Network," *International Journal of Advanced Science and Technology*, vol. 3, pp. 69–78, 2009.

[37] G. Loukas, S. Timotheou, and E. Gelenbe, "Robotic wireless network connection of civilians for emergency response operations," in *2008 23rd International Symposium on Computer and Information Sciences*, 2008, pp. 1–6. [Online]. Available: http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=4717943

[38] Y.-c. Wang, W.-c. Peng, M.-h. Chang, and Y.-c. Tseng, "Exploring Load-Balance to Dispatch Mobile Sensors in Wireless Sensor Networks," in *2007 16th International Conference on Computer Communications and Networks*, 2007, pp. 669–674. [Online]. Available: http://ieeexplore.ieee.org/articleDetails.jsp?arnumber=4317894

[39] U. Baroudi, G. Sallam, M. Al-Shaboti, and M. Younis, "GPS-free robots deployment technique for rescue operation based on landmark's criticality," *2015 International Wireless Communications and Mobile Computing Conference (IWCMC)*, 2015.

[40] "PID Controller [Online]. Available: http://en.wikipedia.org/wiki/PID_controller."

[41] "Sim.I.am: A Robot Simulator by Jean-Pierre de la Croix."

[42] "Khepera 3 [Online]. Available: http://www.k-team.com/mobile-robotics-products/khepera-iii."

Vitae

- Name: Gamal Ameen Sallam
- Nationality: Yemen
- Date of Birth: January 6, 1986
- Email: *gamal_sallam1989@yahoo.com*
- Permenant Address: Ibb, Yemen
- Publications:

  1. G. Sallam, U. Baroudi, COVER: A Cooperative Virtual Force Robot Deployment Technique, The 14th IEEE International Conference on Ubiquitous Computing and Communications IUCC 2015), October 2015.
  2. U. Baroudi, G. Sallam, M. Al-shaboti, M. Younis, GPS-Free Robots Deployment Technique for Rescue Operation Based on Landmarks Criticality, The 11th International Wireless Communications & Mobile Computing Conference (IWCMC 2015), August 2015.
  3. G. Sallam, U. Baroudi, M. Alshaboti, Robots Deployment Using Virtual Force Approach: Challenges and Guidelines, Special Issue Wireless Sensor and Actuator Networks, Electronics 2016, submitted.

4. G.Sallam, U. Baroudi, Real implementation of COVER: A Cooperative Virtual Force Robot Deployment Technique, Under preperation.
5. G.Sallam, U. Baroudi, A Cooperative Virtual Force Robot Deployment Technique to achieve maximal demand satisfaction, Under preperation.